ระบบสารสนเทศแผนที่เครือข่ายคอมพิวเตอร์
COMPUTER NETWORK MAP INFORMATION SYSTEM

โดย

นางสาวกฤษณา บึงราษฎร์

นางสาววลัยลักษณ์ สุวรรณพงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

หัวข้อปริญญานิพนธ์	ระบบสารสนเทศแผนที่เครือข่ายคอมพิวเตอร์
TITLE	**COMPUTER NETWORK MAP INFORMATION SYSTEM**
โดย	นางสาวกฤษณา บึงราษฎร์ รหัสประจำตัว 41014013
	นางสาววลัยลักษณ์ สุวรรณพงศ์ รหัสประจำตัว 41014375
อาจารย์ที่ปรึกษา	อาจารย์นภพินทุ์ อนันตรศิริชัย
	อาจารย์ภุชงค์ หงษ์สุวรรณ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2544

ปริญญานิพนธ์ฉบับนี้ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

.....

(อาจารย์ภุชงค์ หงษ์สุวรรณ)
อาจารย์ผู้ควบคุมปริญญานิพนธ์

หัวข้อวิทยานิพนธ์	ระบบสารสนเทศแผนที่เครือข่ายคอมพิวเตอร์ COMPUTER NETWORK MAP INFORMATION SYSTEM
นักศึกษา	นางสาวกฤษณา บึงราษฎร์ รหัสประจำตัว 41014013 นางสาววลัยลักษณ์ สุวรรณพงศ์ รหัสประจำตัว 41014375
อาจารย์ที่ปรึกษา	อาจารย์นภพินทุ์ อนันตรศิริชัย อาจารย์ภุชงค์ หงษ์สุวรรณ
ระดับการศึกษา	ปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

ระบบสารสนเทศแผนที่เครือข่ายคอมพิวเตอร์ถูกพัฒนาขึ้นโดยใช้ภาษาเชิงวัตถุ และเทคโนโลยีจาวา เพื่อแสดงโครงสร้าง และสถานะของระบบเครือข่ายคอมพิวเตอร์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ณ เวลาปัจจุบันในรูปแบบกราฟฟิก เพื่อช่วยให้ผู้ดูแลระบบเครือข่ายสามารถตรวจสอบสถานะของอุปกรณ์ต่างๆ ในระบบ หรือเก็บข้อมูลต่างๆในระบบได้อย่างมีประสิทธิภาพ

THESIS TITLE	**COMPUTER NETWORK MAP INFORMATION SYSTEM**		
NAME	**Miss Krissana**	**Buengras**	**No. 41014013**
	Miss Walailak	**Suwanpong**	**No. 41014375**
ADVISOR	**Ms. Noppin**	**Anantarasirichai**	
	Mr. Puchong	**Hongsuwan**	
COURSE	**Bachelor Degree of Information Engineering**		
DEPARTMENT	**Information Engineering**		
YEAR	**2001**		

ABSTRACT

Computer Network Map Information System is an application software developed using UML (Unified Modeling Language) and Java Technology Programming. This application software presents KMITL computer network topology and status in real time and graphically in order to support the KMITL network administrator in monitoring and checking the status of all network devices , or correcting the information in the system in the effective way.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดีด้วยความช่วยเหลือเป็นอย่างดีของอาจารย์ทั้งสองท่านนี้ คือ อาจารย์ นภพินทุ์ อนันตรศิริชัย และอาจารย์ภุชงค์ หงษ์สุวรรณ ซึ่งท่านเป็นอาจารย์ที่ปรึกษาและให้คำแนะนำในการจัดทำโครงงาน

ขอกราบขอบพระคุณท่านอาจารย์ทุกท่านที่ได้ให้ความรู้ในด้านวิชาการ ทั้งแนวคิดในการทำงาน และแนวคิดในการดำเนินชีวิต ขอขอบคุณเพื่อนๆ ที่ให้กำลังใจและคำแนะนำดี ๆ พร้อมทั้งขอขอบพระคุณ ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สุดท้ายนี้ขอขอบพระคุณบิดรมารดาที่ได้ให้การสนับสนุนในเรื่องต่างๆ ตลอดมา

นางสาวกฤษณา บึงราษฎร์
นางสาววลัยลักษณ์ สุวรรณพงศ์

ผู้จัดทำ

สารบัญ

สารบัญ

(ต่อ)

สารบัญ

(ต่อ)

สารบัญรูป

สารบัญรูป

(ต่อ)

สารบัญตาราง

บทที่ 1
บทนำ

1.1 แนวคิดและที่มา

เนื่องจากระบบเครือข่ายคอมพิวเตอร์ในปัจจุบันมีข้อมูลต่าง ๆ ที่มีความสำคัญต่อการทำงานของผู้ดูแลระบบอยู่เป็นจำนวนมาก หลากหลาย และมีการเปลี่ยนแปลงอยู่ตลอดเวลาตามสภาพการใช้งาน ดังนั้นการเก็บรวบรวมข้อมูลต่าง ๆ และสร้างเป็นระบบสารสนเทศของเครือข่ายคอมพิวเตอร์ที่มีการจัดการกับข้อมูล และแสดงผลข้อมูลอย่างเหมาะสมตรงตามสภาพการทำงานจริงจึงมีความจำเป็นอย่างยิ่งในการดูแลระบบให้เกิดประสิทธิภาพสูงสุด

ดังนั้นปริญญานิพนธ์นี้จึงนำเสนอการออกแบบ และการสร้างระบบสารสนเทศเพื่อเก็บรวบรวมข้อมูลต่าง ๆ ของระบบเครือข่ายคอมพิวเตอร์ ซึ่งเน้นการแสดงผลของข้อมูลในลักษณะของแผนที่ระบบเครือข่ายคอมพิวเตอร์ในแบบกราฟฟิก ณ เวลาปัจจุบัน เพื่อช่วยการทำงานของผู้ดูแลระบบเครือข่ายให้สามารถตรวจสอบ ดูแลอุปกรณ์ต่าง ๆ ในระบบได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์

1. เพื่อพัฒนาแอปพลิเคชั่นที่ใช้งานทางด้านระบบเครือข่ายคอมพิวเตอร์
2. เพื่อศึกษาการพัฒนาแอปพลิเคชั่นโดยใช้ภาษาจาวา (Java Language)
3. เพื่อศึกษาการออกแบบระบบโดยใช้วิธีการของ UML (Unified Modeling Language)
4. เพื่อศึกษาระบบเครือข่ายคอมพิวเตอร์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

1.3 ผลที่คาดว่าจะได้รับ

1. สามารถสร้างโปรแกรมช่วยการทำงานของผู้ดูแลระบบเครือข่ายคอมพิวเตอร์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ให้เหมาะสม เกิดประสิทธิภาพ และตรงตามความต้องการของผู้ดูแลระบบเครือข่ายมากที่สุด
2. ได้รับความรู้ในด้านระบบเครือข่ายคอมพิวเตอร์ ทั้งในด้านองค์ประกอบ อุปกรณ์ การทำงานของระบบ และการดูแลระบบเครือข่าย
3. ได้รับความรู้ในการออกแบบ และพัฒนาแอปพลิเคชั่น

1.4 ขอบเขตของปัญหา

1. เครื่องมือที่ใช้ในการพัฒนาระบบ
    - Java Technology, Java Programming Language : เป็นเทคโนโลยีในการพัฒนาระบบที่ดีเพราะมีความสามารถที่โดดเด่นในเรื่องต่าง ๆ ดังนี้ Object-Oriented programming , networking , security , multithreaded programming , distributed , high performance
    - Object-Oriented Technology : เป็นวิธีการที่ใช้ในการวิเคราะห์ปัญหา และแก้ปัญหาเพราะมองปัญหาและองค์ประกอบต่าง ๆ เป็นวัตถุ ซึ่งคุณสมบัติที่เด่นของวัตถุมีดังนี้ การแยกแยะเอกลักษณ์ (Abstraction) , การสืบทอดคุณสมบัติ (Inheritance) , โพลีมอร์พิซึม (Polymorphism) , การซ่อนรายละเอียด (Encapsulation) , การส่งแมสแสจ (Message Sending) , การเป็นส่วนหนึ่งของ (Aggregation) , ความสัมพันธ์แบบเอซโซซิเอชัน (Associations)
2. ความสามารถและคุณสมบัติของระบบ

    สามารถแสดงแผนที่เครือข่ายคอมพิวเตอร์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังในรูปแบบกราฟฟิก และแสดงรายละเอียดต่าง ๆ ของระบบได้ดังต่อไปนี้
    - แสดงอุปกรณ์หลักของเครือข่ายดังนี้ FDDI ring, concentrators, routers, computers, links
    - แสดงการเชื่อมต่อของอุปกรณ์ต่าง ๆ ในระบบ
    - แสดงสถานะของอุปกรณ์ในปัจจุบันว่าเปิดหรือปิด ใช้การได้หรือเสียหาย
    - แสดงสถานที่ตั้งและ เจ้าหน้าที่ผู้ดูแลอุปกรณ์
    - แสดงหมายเลย IP address ของอุปกรณ์ต่าง ๆ ในระบบ
    - แสดงบริการต่าง ๆ ที่เครื่องคอมพิวเตอร์เครื่องหนึ่งให้บริการอยู่ หรือใช้บริการอยู่
3. ฮาร์ดแวร์ (Hard ware) ที่ใช้ในระบบ
    - เครื่องคอมพิวเตอร์ระบบปฏิบัติการ Microsoft Windows ที่เชื่อมต่อกับระบบเครือข่ายคอมพิวเตอร์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

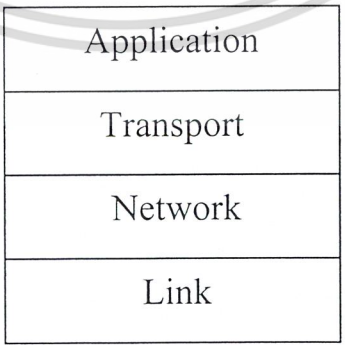
บทที่ 2
โปรโตคอลทีซีพี / ไอพี (TCP/IP), ไอซีเอ็มพี (ICMP) และการพัฒนาระบบเชิงวัตถุ

โปรโตคอลทีซีพี/ไอพี เป็นชุดของโปรโตคอลที่มีการพัฒนามาตั้งแต่ปี 1960 โดยมีวัตถุประสงค์ให้สามารถสื่อสารจากต้นทางข้ามเน็ทเวิร์กไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปเองได้โดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเน็ทเวิร์กที่มีปัญหา โปรโตคอลก็ยังสามารถหาเส้นทางส่งผ่านข้อมูลไปยังปลายทางได้ ในระยะเริ่มต้นโปรโตคอลนี้ใช้กันในวงแคบๆ จนในช่วงปี 90 จึงมีการนำมาใช้ในทางธุรกิจ และเป็นจุดเริ่มต้นของอินเทอร์เน็ตในปัจจุบัน จากนั้นมีการปรับปรุงแก้ไขข้อบกพร่องต่างๆ จนเป็นโปรโตคอลที่มีการใช้อย่างแพร่หลายในปัจจุบันไม่ว่าจะเป็นเครื่อข่ายเฉพาะหรือเครือข่ายในวงกว้าง ซึ่ง ทีซีพี/ไอพี (TCP/IP) นั้นจะเชื่อมกลุ่มเครือข่ายย่อยเข้าด้วยกันเป็นเครือข่ายใหญ่หรือ อินเทอร์เน็ต (internet)

ทีซีพี/ไอพีนั้นผ่านการออกแบบให้เป็นอิสระจากชนิดของคอมพิวเตอร์ ฮาร์ดแวร์ (hardware) และระบบปฏิบัติการ(Operating System) กลไกของโปรโตคอลนั้นมีความเชื่อถือได้สูงและทำงานได้แม้ในบางสภาวะที่การสื่อสารมีความผิดปกติ รวมทั้งสามารถเลือกเส้นทางส่งข้อมูลตามสภาพเครือข่ายได้ในกรณีบางเส้นทางชำรุด

2.1 การแบ่งชั้น (Layering)

ทีซีพี/ไอพี เป็นชุดของโปรโตคอลที่ประกอบด้วยโปรโตคอลย่อยหลายตัวโดยแต่ละตัวจะทำหน้าที่ในแต่ละชั้นหรือเลเยอร์ (layer) ซึ่งรับผิดชอบและแปลความหมายของข้อมูลในแต่ละระดับของการสื่อสาร ซึ่งภาพรวมแล้วทีซีพี/ไอพี แบ่งออกเป็น 4 เลเยอร์ดังรูป



รูปที่ 2.1 แสดงระดับของทีซีพี/ไอพี

หน้าที่ความรับผิดชอบแต่ละเลเยอร์มีดังนี้

1. ลิงค์เลเยอร์ (Link Layer) ในเลเยอร์นี้จะเป็นดีไวซ์ไดรเวอร์ที่ทำงานอยู่บนระบบปฏิบัติการแต่ละระบบทำหน้าที่รับผิดชอบในการรับส่งข้อมูลตั้งแต่ระดับกายภาพ, สัญญาณไฟฟ้า จนถึงการแปลความจากระดับสัญญาณไฟฟ้าจนเป็นข้อมูลทางคอมพิวเตอร์, โปรโตคอลระดับนี้ เช่น อีเทอร์เน็ต (Ethernet) และ SLIP (Serial Line Internet Protocol)
2. เน็ทเวิร์กเลเยอร์ (Network Layer) รับผิดชอบในการรับ-ส่งข้อมูลไปจนถึงจุดหมายปลายทาง โปรโตคอลระดับนี้ได้แก่ IP, ICMP, IGMP
3. ทรานสปอร์ตเลเยอร์ (Transport Layer) รับผิดชอบในการรับส่งข้อมูลระหว่างเครื่องหนึ่ง (host) ไปยังอีกโฮสต์หนึ่ง และจะส่งข้อมูลขึ้นไปให้แอพพลิเคชันเลเยอร์ (Application Layer) นำไปใช้งานต่อ มีโปรโตคอลที่จัดอยู่ในเลเยอร์นี้คือ ทีซีพี และ ยูดีพี (UDP) ซึ่งมีลักษณะของการรับส่งข้อมูลที่แตกต่างกันออกไป
4. แอพพลิเคชันเลเยอร์ (Application Layer) เป็นเลเยอร์ที่แอพพลิเคชันเรียกใช้โปรโตคอลระดับล่างๆ ลงไป เพื่อวัตถุประสงค์แตกต่างกัน เช่น

FTP (File Transfer Protocol)	ใช้สำหรับรับส่งแฟ้มข้อมูลระหว่างโฮสต์
SMTP (Simple Mail Transfer Protocol)	ใช้รับส่งจอหมายอิเล็กทรอนิกส์ระหว่างโฮสต์
Telnet	ใช้สำหรับควบคุมเครื่องระยะไกล
HTTP (Hypertext Transfer Protocol)	เป็นโปรโตคอลที่ใช้รับส่งข้อมูลเว็บเพจระหว่างบราวเซอร์และเว็บเซิร์พเวอร์
POP (Post Office Protocol)	ใช้สำหรับดาวน์โหลดอีเมล์จากเมล์เซิร์ฟเวอร์มาไว้ที่เครื่องเมล์ไคลเอนท์ (PC) ของผู้ใช้

ทีซีพี (TCP) เป็นโปรโตคอลที่รับประกันการรับ-ส่งข้อมูลระหว่างโฮสต์ กล่าวคือ โปรโตคอลมีกลไกในการตรวจสอบและยืนยันว่ามีข้อมูลจากต้นทางจะไปถึงปลายทางเสมอ หากข้อมูลถึงปลายทางก็จะมีสัญญาณตอบรับว่าข้อมูลถึงปลายทางแล้ว หากไม่มีสัญญาณตอบรับก็แสดงว่าข้อมูลไม่ถึงปลายทาง ดังนั้นแอพพลิเคชันที่มีความสำคัญจึงเลือกใช้โปรโตคอลนี้ในการรับส่งข้อมูล กระบวนการยืนยันการรับส่งข้อมูลนี้เองเป็นจุดเด่นของทีซีพี

ยูดีพี (UDP) เป็นโปรโตคอลแบบ คอนเน็คชั่นเลส (connectionless) คือ ไม่ต้องสถาปนาการเชื่อมต่อระหว่างสถานีรับและสถานีส่ง ยูดีพีนั้นเป็นโปรโตคอลระดับชั้นเดียวกันกับทีซีพีแต่ว่าไม่มีกลไกในการรับประกันความเชื่อถือในการขนถ่ายข้อมูล หากข้อมูลสูญหาย ซ้ำซ้อน หรือลำดับไม่ถูกต้อง ยูดีพีจะปล่อยให้โปรโตคอลที่เรียกใช้งานดำเนินการกับปัญหาเหล่านี้เอง ซึ่ง

การเลือกใช้โปรโตคอลนี้จะขึ้นอยู่กับแอพพลิเคชันเป็นผู้ดำเนินการตรวจสอบยืนยันเอง หากว่าต้องการตรวจสอบ
ก็ให้รับส่งข้อมูลยืนยันเพิ่มเติม ที่เป็นเช่นนี้ก็เพราะว่าบางแอพพลิเคชันนั้นไม่จำเป็นที่ต้องมีการรับประกันข้อมูล อาจจะขอเพียงให้สามารถรับ- ส่งข้อมูลได้อย่างมีประสิทธิภาพก็เพียงพอแล้ว

2.1.2 การแบ่งชั้นทีซีพี/ไอพี

ในชุดของโปรโตคอลทีซีพี/ไอพี ประกอบด้วยโปรโตคอลหลายตัวทำงานร่วมกันในเลเยอร์ต่างๆ และมีหน้าที่แตกต่างกันไป ดังภาพ 2.2 แสดงถึงโปรโตคอลในแต่ละเลเยอร์ที่เมื่อรวมกันเป็นชุดของ ทีซีพี/ไอพี ก็คือ

TCP : อยู่ในทรานสปอร์ตเลเยอร์ ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูลให้มีเสถียรภาพและเชื่อถือได้

UDP : อยู่ในทรานสปอร์ตเลเยอร์ ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูลเช่นเดียวกัน แต่ไม่มีกลไกในการรับส่งที่มีเสถียรภาพและเชื่อถือได้ โดยปล่อยหน้าที่นี้ให้กับแอพพลิเคชันเลเยอร์เป็นผู้ทำหน้าที่นี้แทน

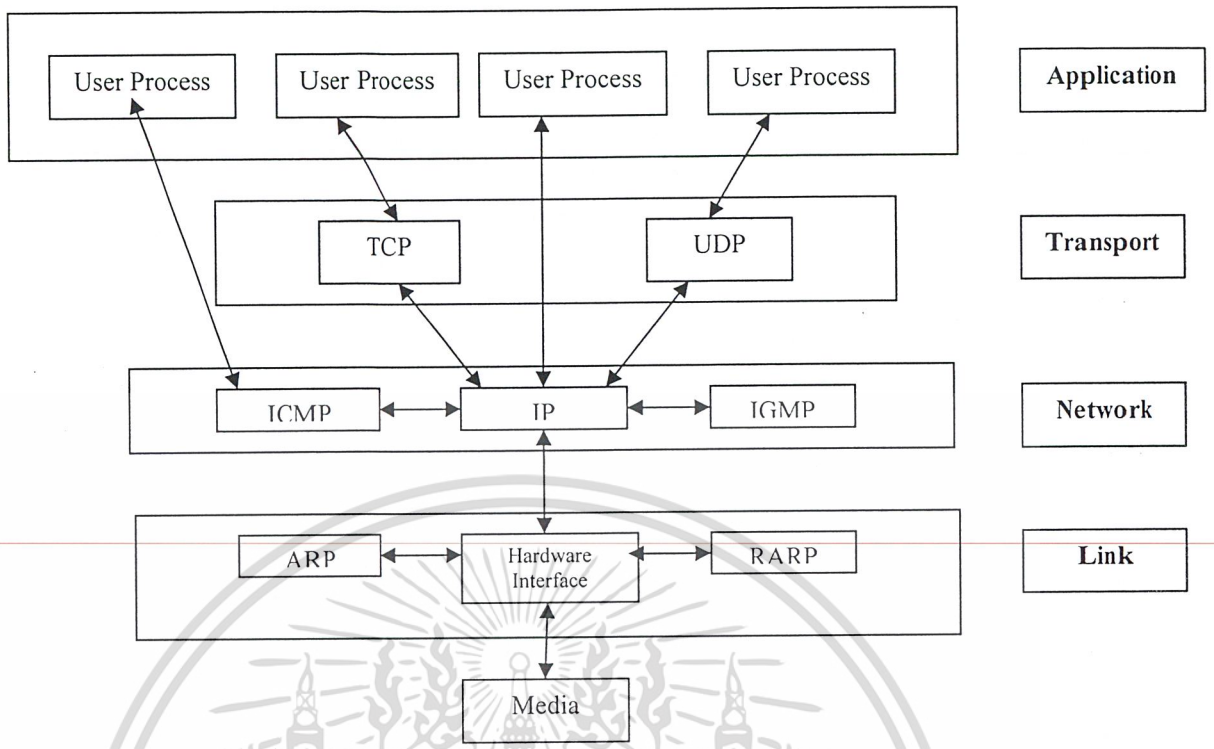
IP : อยู่ในเน็ทเวิร์กเลเยอร์ เป็นโปรโตคอลหลักในการสื่อสารข้อมูล ซึ่งกลไกสำคัญที่ทำให้ข้อมูลสามารถเคลื่อนที่ไปยังปลายทางได้คือ โปรโตคอลไอพีนี้เอง

ICMP: Internet Control Message Protocol อยู่ในเน็ทเวิร์กเลเยอร์ ทำหน้าที่เสริมให้การทำงานของไอพีให้สมบูรณ์ โดยจะเป็นโปรโตคอลที่คอยส่งข่าวสารและการแจ้งความผิดพลาดให้แก่ไอพี แต่ในบางโอกาสแอพพลิเคชันเลเยอร์ก็เรียกใช้ ICMP โดยตรงเพื่อใช้ประโยชน์จากความสามารถของ ICMP ด้วยเช่นกัน

IGMP: Internet Group Message Protocol อยู่ในเน็ทเวิร์กเลเยอร์ ทำหน้าที่ในการส่ง ยูดีพี เดทาแกรมไปยังกลุ่มของโฮสต์ หรือ โฮสต์หลายๆ ตัวพร้อมกัน

ARP: Address Reservation Protocol อยู่ในลิงค์เลเยอร์ทำหน้าที่เปลี่ยนระหว่างแอดเดรสที่ใช้โดยไอพีให้เป็นแอดเดรสของการติดต่อกับเน็ทเวิร์ก (Network Interface)

RARP: Reverse ARP อยู่ในลิงค์เลเยอร์เช่นกัน แต่ทำหน้าที่กลับกันกับ ARP คือ เปลี่ยนระหว่างแอดเดรสของการติดต่อกับเน็ทเวิร์ก ให้เป็นแอดเดรสที่ใช้โดย ไอพี



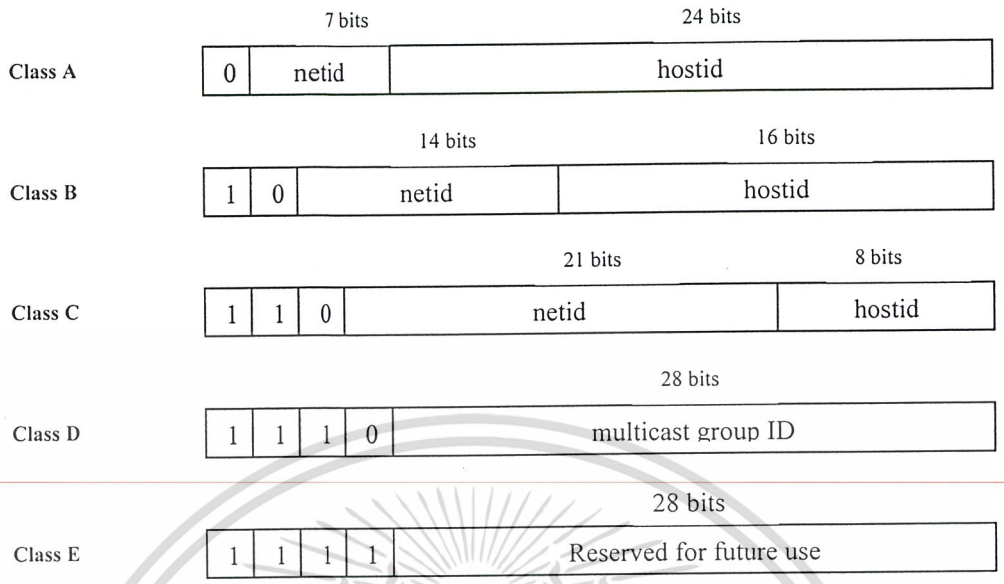
รูปที่ 2.2 เลเยอร์ของโปรโตคอลต่างๆ ในชุดของ ทีซีพี/ไอพี

2.1.3 อินเทอร์เน็ทแอดเดรส (Internet Address)

ทุกอินเตอร์เฟซที่ต่ออยู่ในอินเตอร์เน็ทจะต้องมีหมายเลขประจำตัวเพื่อใช้ในการสื่อสารข้อมูลเรียกว่า อินเทอร์เน็ทแอดเดรส (Internet Address) หรือเรียกย่อๆ ว่า ไอพีแอดเดรส (IP Address) โดยค่านี้จะเป็นหมายเลขจำนวน 32 บิท แต่แทนที่จะกำหนดให้เลขทั้ง 32 บิทนั้นถูกนับต่อเนื่องกันไป ตั้งแต่ $0 - 2^{32}$ ก็จะใช้วิธีการแบ่งหมายเลขดังกล่าวออกเป็นกลุ่มของเลขขนาด 8 บิท จำนวน 4 ชุด และคั่นแต่ละชุดด้วยจุด ตัวอย่างเช่น 161.246.48.11

นอกจากนี้ใน ไอพีแอดเดรส นั้นยังถูกแบ่งออกเป็น 2 ส่วน คือ ส่วนที่เป็นแอดเดรสของเน็ตเวิร์ก (Network ID) และส่วนที่เป็นแอดเดรสของโฮสต์ (Host ID) ซึ่งข้อมูลในส่วนนี้จะถูกใช้สำหรับค้นหาเส้นทางของไอพี ในการที่จะขนส่งข้อมูลจากต้นให้ถึงปลายทางอย่างถูกต้อง

เพื่อเป็นการกำหนดขนาดของเน็ตเวิร์กสำหรับไอพีแอดเดรส (IP Address) ต่างๆ ดังนั้นจึงมีการจัดการไอพีแอดเดรสในแต่ละออกเป็นคลาส (class) ต่างๆกันจาก A ถึง E เพื่อจะได้ทำการจัดสรรไอพีแอดเดรสได้อย่างเหมาะสมกับขนาดของเน็ตเวิร์ก



รูปที่ 2.3 ก การกำหนดแอดเดรสสำหรับคลาสต่างๆ

Class	Range
A	0.0.0.0 – 127.255.255.255
B	128.0.0.0 – 191.255.255.255
C	192.0.0.0 – 223.255.255.255
D	224.0.0.0 – 239.255.255.255
E	240.0.0.0 – 255.255.255.255

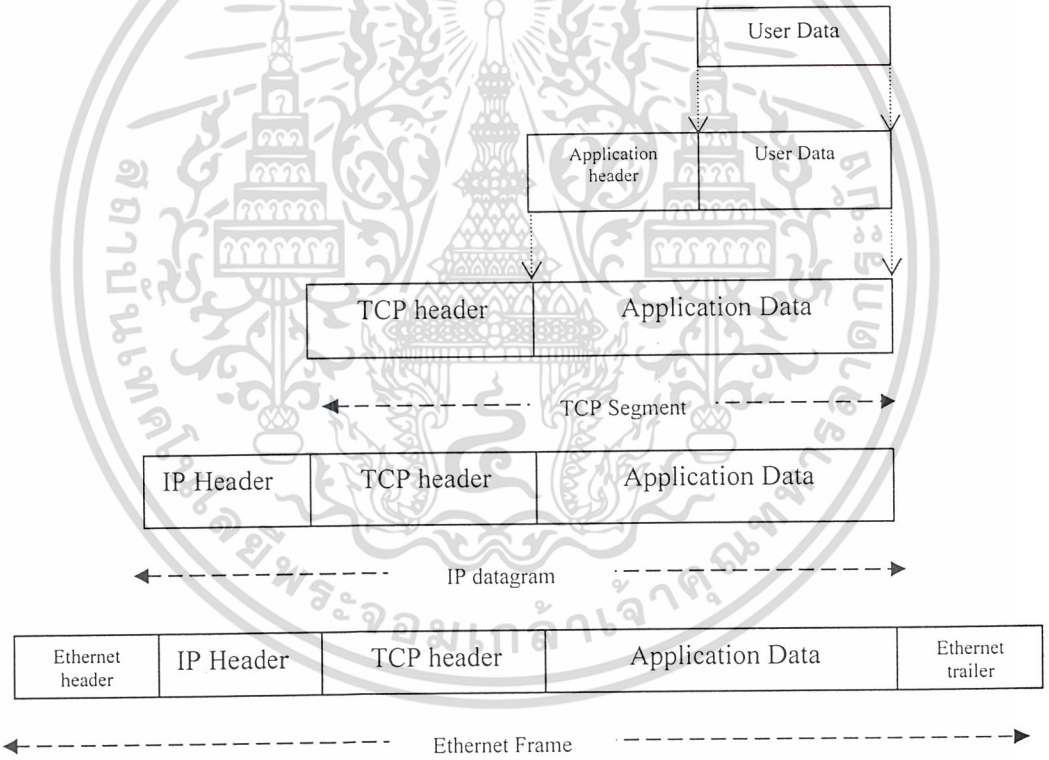
รูปที่ 2.3 ข แสดงช่วงไอพีแอดเดรส แต่ละคลาส

2.1.4 การซ่อนรายละเอียด (Encapsulation)

การซ่อนรายละเอียดคือการนำข้อมูลที่ต้องการส่งมาประกอบรวมกับข้อมูลที่เป็นส่วนควบคุมของโปรโตคอล โดยข้อมูลส่วนที่เป็นส่วนควบคุมนั้นจะถูกนำมาไว้ในส่วนหัวของข้อมูล เรียกว่า

เฮดเดอร์ (header) ซึ่งใช้ในการรับข้อมูลนั้นผู้ที่รับข้อมูลจะได้รับเฮดเดอร์ก่อนจากนั้นก็นำเฮดเดอร์ไปแปลและทราบว่าข้อมูลที่ตามมานั้นมีลักษณะอย่างไรจะได้จัดการได้อย่างถูกต้อง

ภายในเฮดเดอร์ของโปรโตคอลส่วนใหญ่จะประกอบด้วยข้อมูลหลักที่สำคัญของโปรโตคอลที่ทำ การซ่อนรายละเอียดมาคือแอดเดรสต้นทาง, แอดเดรสปลายทาง, ความยาวข้อมูล, รหัสตรวจความผิดพลาดข้อมูล ซึ่งสิ่งที่จะต้องเน้นให้เห็นชัดก็คือ จะมีข้อมูลสำคัญเฉพาะโปรโตคอลที่ทำการซ่อนรายละเอียดมาเท่านั้น ตัวอย่างเช่น การซ่อนรายละเอียดของอีเทอร์เน็ท (Ethernet) ก็จะมีการระบุอีเทอร์เน็ทแอดเดรส(Ethernet address) ลงในเฮดเดอร์เท่านั้น จะไม่มีการบรรจุไอพีแอดเดรส ลงมาในอีเทอร์เน็ทเฮ็ดเดอร์(Ethernet Header) ด้วยแต่อย่างใด เพราะในเลเยอร์ของอีเทอร์เน็ทจะไม่รู้จัก ไอพีแอดเดรส หรือ รหัสควบคุมใดๆ ของไอพีจะถูกตีค่าว่าเป็นข้อมูลก้อนเดียวกันสำหรับ อีเทอร์เน็ทเท่านั้น



รูปที่ 2.4 การซ่อนรายละเอียดข้อมูลผ่านชั้นของโปรโตคอลแต่ละระดับ

ในการรับส่งข้อมูลนั้น ข้อมูลที่รับส่งกับจริงๆ บนเน็ทเวิร์กนั้นจะประกอบไปด้วยสองส่วน คือ ข้อมูลจริงกับข้อมูลของโปรโตคอล เปรียบเสมือนการส่งจดหมายซึ่งจะต้องประกอบด้วยเนื้อความในจดหมายและซองจดหมายที่เขียนชื่อที่อยู่ ติดแสตมป์ ถ้ามีแต่จดหมายอย่างเดียว

ไปรษณีย์ก็ต้องอ่านเองเพราะไม่รู้ว่าเราจะส่งให้ใคร การซ่อนรายละเอียดก็คือ การเอาจดหมายมาใส่ซองนั่นเอง โดยซองจะเปรียบเหมือนข้อมูลที่ใช้ในการรับส่งข้อมูลของ โปรโตคอลนั้น 1 โปรโตคอลก็จะใส่ 1 ซอง ถ้าข้อมูลต้องส่งผ่านหลายเลเยอร์ จำนวนซองก็จะถูกใส่เพิ่มหลายชั้นตามลำดับการซ่อนรายละเอียดนั่นเอง ดังนั้นเราจะส่งข้อมูลผ่านโปรโตคอลทีซีพี ข้อมูลเราเตอร์ก็จะถูกใส่ซองตามลำดับดังนี้

ลำดับที่ 1 ซอง ทีซีพี
ลำดับที 2 ซอง ไอพี
ลำดับที่ 3 ซอง อีเทอร์เน็ท

และฝ่ายที่รับข้อมูลก็จะต้องแกะซองออกตามลำดับ โดยจะต้องแกะซองอีเทอร์เน็ทก่อนแล้วจะเจอซองไอพี แกะซองไอพีจะเจอซองทีซีพี แกะซองทีซีพีก็จะเจอข้อมูลที่ต้องการ

การซ่อนรายละเอียดในแต่ละระดับก็จะมีการเรียกข้อมูลที่อยู่ในซองแตกต่างกันออกไป ข้อมูลที่ทำการซ่อนรายละเอียด เรียบร้อย (ใส่ซองปิดผนึก) แล้วจากทีซีพีส่งไปยังไอพี เรียกว่า ทีซีพีเซกเมนต์ (TCP Segment)

ในระดับไอพี ก็จะถือว่าทีซีพีเซกเมนต์ เป็นข้อมูลทั้งชุด เมื่อไปรวมกับไอพีเฮ็ดเดอร์ (IP Header) ส่งไปยังเลเยอร์ดาตาลิงค์ (Data link) จะเรียกว่าไอพีดาตาแกรม (IP Datagram)

จากรูปจะเห็นว่าบางครั้งข้อมูลเรามีอยู่เล็กน้อย แต่กว่าที่เราจะส่งข้อมูลไปถึงปลายทางได้จะมีข้อมูลของเฮดเดอร์ของโปรโตคอลเกาะติดไปด้วยเสมอ เช่นเดียวกับที่บางครั้งเราได้รับของขวัญที่กล่องใหญ่ห่อหลายชั้น แต่พอเปิดเข้าไปข้างในอาจเป็นเพียงของชิ้นเล็กๆ เพียงชิ้นเดียว เป็นต้น ซึ่งก็เป็นไปได้ บางครั้งอาจจะดูสิ้นเปลืองแต่การที่ต้องใส่ซองหลายชั้นแล้วส่งถึงที่หมายดีกว่าประหยัดซองแต่จดหมายไม่ถึง สำหรับในการสื่อสารข้อมูลจริงๆ ไม่มีซองให้เปลือง แต่ทรัพยากรที่เราสิ้นเปลืองไปก็คือแบนด์วิดที่เราอาจใช้งานได้ไม่เต็มประสิทธิภาพเท่าที่ควร เนื่องจากทุกแพ็กเก็ตของข้อมูลจะต้องเสียส่วนหนึ่งไปเป็นเฮ็ดเดอร์เสมอ

2.1.5 การดีมัลติแพลกซ์ (De multiplexing)

ในหัวข้อที่ผ่านมาได้กล่าวถึงการซ่อนรายละเอียด ไปแล้วการดีมัลติแพลกซ์ คือกระบวนการย้อนหลังของการซ่อนรายละเอียด นั่นเอง หากกระบวนการซ่อนรายละเอียดคือการนำข้อมูลมาใส่ซองทีละชั้นตามเลเยอร์ที่ส่งไป การดีมัลติแพลกซ์ ก็คือ การรับซองข้อมูลที่ปิดผนึกใส่ซองมาอย่างมิดชิด เพื่อทำการแกะออกทีละชั้นตามเลเยอร์จนถึงเลเยอร์บนสุดคือแอพพลิเคชั่นเลเยอร์ จึงได้ข้อมูลเนื้อความจริงๆ ที่ต้องการสื่อสารกัน

2.2 ความสำคัญของหมายเลขพอร์ต (Port Number)

จากเลเยอร์ที่แสดงในภาพที่ 2.1 จะเห็นว่าเลเยอร์บนสุดของทีซีพี/ไอพี คือแอพพลิเคชั่นเลเยอร์ สิ่งหนึ่งที่เราสามารถสังเกตเห็นได้ว่าข้อมูลทั้งหมดทุกๆ เซกเมนต์จะต้องผ่านเลเยอร์นี้ก็เพราะแอพพลิเคชั่นเลเยอร์จะครอบคลุมทั้งหมด ดังนั้นจะเกิดอะไรขึ้นถ้ามีหลายแอพพลิเคชั่นต่างก็ต้องการรับส่งข้อมูลผ่านทีซีพี/ไอพี แต่ละแอพพลิเคชั่นจะสามารถแยกแยะได้อย่างไร ในความเป็นจริงที่ใช้งานในปัจจุบันก็มีอยู่มากที่มีแอพพลิเคชั่นมากกว่า 1 แอพพลิเคชั่นที่ทำงานอยู่ภายในเซิร์พเวอร์นอกจากนี้อาจจะมีการบริการอื่นๆ บนทีซีพี/ไอพี ที่ซ่อนอยู่โดยที่เราไม่อาจทราบ

พอร์ต (port) จะเป็นปัญหาของคำถามข้างต้น ในโปรโตคอลทีซีพี/ไอพี ได้ออกแบบให้มีหมายเลขพอร์ตอยู่ในเฮดเดอร์ เพื่อระบุว่าเซกเมนต์นี้เป็นของแอพพลิเคชั่นอะไร ในโฮสต์นั้นแอพพลิเคชั่นแต่ละตัวที่ให้บริการอยู่ในเครื่องต่างมีหมายเลขพอร์ตประจำตัว เพื่อจะสามารถเลือกนำข้อมูลมาใช้ว่าเป็นแอพพลิเคชั่นของตัวเองหรือไม่ หมายเลขพอร์ตที่เรารู้จักกันดีและเป็นมาตรฐาน ได้แก่ พอร์ต 20, 21 เป็นพอร์ตของ FTP , พอร์ต 25 SMTP , พอร์ต 30 HTTP เป็นต้น

โดยทั่วไปหมายเลขพอร์ตจะมีความสำคัญกับฝั่งของเซิร์ฟเวอร์เท่านั้น เนื่องจากแอพพลิเคชั่นฝั่งเซิร์ฟเวอร์จะต้องคอยรับการรีเควสท์ (request) หรือขอรับบริการจากไคล์เอนต์ที่พอร์ตเดิมเสมอ ส่วนในฝั่งไคล์เอนต์เองหมายเลขพอร์ตไม่จำเป็นต้องเป็นหมายเลขตายตัวและคงที่ เพราะหมายเลขพอร์ตจะเป็นการสุ่มหมายเลขขึ้นมาใช้ชั่วคราว (Ephemeral Port) และจะมีการเรียกใช้พอร์ตใหม่ทุกครั้งที่มีการรับส่งข้อมูลเซสชั่น (session) ใหม่

การเรียกใช้พอร์ต (Request Port)

ในระบบปฏิบัติการยูนิกซ์ (Unix) มีการสงวนพอร์ตบางส่วนไว้ให้สำหรับโพรเซสที่มีสิทธิพิเศษของซูปเปอร์ยูสเซอร์ (Super user) เท่านั้นที่สามารถใช้ในพอร์ตในช่วง 1 –1023 ได้ แต่สำหรับวินโดว์เอ็นที (Windows NT) มิได้สงวนไว้แต่เพียงอย่างใด ในบางครั้งคำว่ายูนิกซ์รีเวร์สพอร์ต (Unix Reserved Port) ก็คือพอร์ต 1- 1023 นั่นเอง

ตารางที่ 2.1 พอร์ตมาตรฐานของทีซีพี

PORT NUMBER	SERVICES
1	Tcpmux
7	Echo
13	Time
17	Qold (Quote of day)
19	Chargen
21	Ftp
22	Ssh (Secure Shell)
23	Telnet
43	Whois
53	DNS (Domain Name Server)
70	Gopher
79	Finger
80	Http (Web Server)
87	Link
95	Supdup
109-110	POP
111	Portmap
135	Epmap
139	NetBios
143	IMAP
144	News window Sys
443	HttpS (Secure Web Server)
512	Remote Exec
513	Remote Login
514	Remote Shell
515	Printer

ตารางที่ 2.2 พอร์ตมาตรฐานของยูดีพี

PORT NUMBER	SERVICES
7	Echo
13	Time
17	Qotd (Quote Of the day)
19	Chargen
53	DNS
67-68	BootP
69	TFTP
88	Keberos
111	Portmap
137-138	NetBIOS
161-162	SNMP
177	X11 logins
513	Who
514	Syslog
517	Talk
518	Ntalk
2049	NFS
5631	PC Anywhere

2.3 หลักการทำงานในการเลือกหาเส้นทาง

การเลือกหาเส้นทางเป็นหน้าที่สำคัญอย่างหนึ่งของไอพี เราเตอร์อาศัยตารางเส้นทางเพื่อนำส่งดาทาแกรมไปยังเครือข่ายปลายทาง ตารางเส้นทางอาจจะปรับเปลี่ยนไปได้ตามสภาพเครื่อข่ายอัตโนมัติโดยใช้โปรโตคอลเลือกเส้นทาง หรือผู้ดูแลระบบอาจตั้งเส้นทางส่งดาตาแกรมเอง

การเลือกเส้นทางเป็นกระบวนการที่เกิดขึ้นในระดับชั้นที่ 3 หรือระดับเน็ทเวิร์กของแบบจำลองทีซีพี/ไอพี ซอร์ฟแวร์ไอพีที่อยู่ในโฮสต์หรือเราท์เตอร์จะนำส่งดาตาแกรมไปตาม

เส้นทางโดยอาศัยเลขเครือข่ายของไอพีแอดเดรสตามแต่ละคลาสเลขเครือข่ายที่เป็นเสมือนค่ากำหนดตำแหน่งปลายทางของเครือข่ายซึ่งคล้ายกับรหัสสามตัวแรกที่กำหนดชุมสายโทรศัพท์

เครือข่ายโดยทั่วไปประกอบด้วยสถานีย่อยปลายทาง (end node) และเราเตอร์ (router) หรืออุปกรณ์อื่นทำงานร่วมกัน กระบวนการเลือกหาเส้นทางจะเกิดขึ้นทั้งที่สถานีส่วนปลายและที่เราเตอร์ ซึ่งสถานีต้นทางเป็นผู้ตัดสินใจขั้นแรกว่าต้องส่งดาทาแกรมไปยังสถานีในเครือข่ายด้วยกันเอง หรือต้องส่งผ่านผ่านเราเตอร์ โดยสถานีต้นทางจะเปรียบเทียบเลขเครือข่ายของแอดเดรสต้นทางและปลายทางกับค่าซับเน็ตมาสค์ หากได้ค่าเครือข่ายที่เหมือนกันแสดงว่าสถานีปลายทางอยู่ในเครือข่ายเดียวกัน สถานีต้นทางจะใช้เออาร์พีสอบถามฮาร์ดแวร์แอดเดรสหรืออ่านจากแคช และบรรจุฮาร์ดแวร์แอดเดรสเข้าสู่แฟรมดาทาลิงค์เพื่อส่งตรงไปยังสถานีปลายทาง แต่ถ้าเลขเครือข่ายมีค่าต่างกันแสดงว่าสถานีปลายทางอยู่ต่างเครือข่ายกัน สถานีส่งจะส่งเฟรมไปให้เราเตอร์เพื่อให้เราเตอร์นำส่งต่อไป เมื่อเราเตอร์ได้รับแฟรมนี้ก็จะส่งต่อไป

ประเภทของการเลือกเส้นทาง

โฮสต์และเราเตอร์จะเก็บแอดเดรสปลายทางสำหรับใช้เป็นเส้นทางในการส่งเดทาแกรมไว้ใน ตารางเส้นทาง (routing table) ซึ่งค่าในตารางจะประกอบด้วยไอพีแอดเดรสของเครือข่าย ปลายทาง และเกตเวย์ เมื่อส่งเดทาแกรมออกไปสู่เน็ทเวิร์กต้องผ่านเราเตอร์จำนวนมาก ซึ่งสามารถเลือกเส้นทางได้หลายเส้นทาง ดังนั้นจึงจำเป็นต้องมีการทำแผนที่เครือข่ายของเราเตอร์คือ ตารางเลือกเส้นทางซึ่งสร้างขึ้นโดยอาศัยวิธีการที่ใช้โดยทั่วไป 2 วิธี คือ

1. การเลือกเส้นทางแบบสเตติก (static routing) ตารางเลือกเส้นทางที่สร้างขึ้นและสามารถแก้ไขโดยผู้ดูแลระบบ
2. การเลือกเส้นทางแบบไดนามิก (dynamic routing) ใช้ซอฟแวร์คำนวณหาค่าตารางเลือกเส้นทาง ตารางจะปรับเปลี่ยนได้หากสภาพเครือข่ายเปลี่ยนไป

2.3.1 การเลือกเส้นทางแบบสเตติก

การเลือกเส้นทางแบบนี้ผู้ที่ดูแลระบบเป็นผู้พิจารณาและคำนวนหาเส้นทางทั้งหมดที่ต้องกำหนดให้เราเตอร์ทุกตัวในเครือข่ายที่ดูแลอยู่และทำตารางเส้นทางให้กับเราเตอร์ซึ่งจะมีค่านี้ตลอดจนกว่าที่ผู้ดูแลระบบทำการเปลี่ยนแปลง ข้อดีของการเลือกเส้นทางแบบสเตติกคือ

1. สะดวกต่อการใช้งานกับเครือข่ายขนาดเล็ก
2. ไม่ต้องใช้เซิร์พเวอร์ในการเลือกเส้นทาง เราเตอร์ไม่ต้องมีซีพียูสมรรถนะสูง
3. ประหยัดแบนด์วิดธ์เครือข่าย เนื่องจากไม่ต้องแลกเปลี่ยนข้อมูลระหว่างเราเตอร์

การเลือกเส้นทางแบบสเตติกนิยมใช้กับการเชื่อมโยงจุดต่อจุดระหว่างเราเตอร์ ตัวอย่างเช่น เครือข่ายที่มีทางออกไปสู่ภายนอกหรืออินเทอร์เน็ทเพียงช่องทางเดียวการกำหนดเส้นทางจะเป็นแบบสเตติก กระนั้นก็ตามการเลือกเส้นทางแบบสเตติกจะมีข้อดีหลายอย่างแต่ว่าก็มีข้อด้อยเช่นกัน

1. ไม่เหมาะกับเครือข่ายขนาดใหญ่ เพราะในการทำตารางแผนที่ของเครือข่ายขนาดใหญ่นั้นมีอุปกรณ์เราเตอร์จำนวนมากยากเกินกว่าที่จะคำนวนและป้อนค่าเข้าสู่เราเตอร์
2. ไม่สะดวกต่อการเปลี่ยนโทโปโลยี เพราะต้องคำนวณและป้อนเส้นทางใหม่
3. เนื่องจากตารางเส้นทางคงตัว ไม่สามารถเปลี่ยนแปลงเองได้ หากเส้นทางใดถูกตัดขาดผู้ดูแลระบบต้องคอยตรวจสอบและแก้ไขปัญหาเอง

2.3.2 การเลือกเส้นทางแบบไดนามิก

การเลือกเส้นทางแบบนี้ต้องใช้ซอฟแวร์ทำหน้าที่แลกเปลี่ยนข้อมูลการเลือกเส้นทางระหว่างเราเตอร์ด้วยกันโดยใช้โปรโตคอลเลือกเส้นทาง (routing protocol) เราเตอร์จะสร้างตารางเลือกเส้นทางจากสภาพเครือข่ายขณะนั้น หากเครือข่ายเปลี่ยนแปลงไปตารางในการเลือกเส้นทางจะเปลี่ยนไปด้วย ข้อดีของการเลือกเส้นทางแบบไดนามิก คือ

1. สามารถรองรับขนาดของเครือข่ายที่ขยายขึ้นได้
2. ตารางเส้นทางเปลี่ยนแปลงค่าเองตามการทำงานของซอฟแวร์ เส้นทางใดที่ถูกตัดขาดจะมีการหาเส้นทางใหม่มาทดแทน

การเลือกเส้นทางแบบไดนามิกนี้ต้องอาศัยการแลกเปลี่ยนค่าเส้นทางระหว่างเราเตอร์และใช้ซีพียูในเราเตอร์เพื่อสร้างตารางเส้นทาง เราเตอร์ประเภทนี้จึงมีราคาสูงกว่าเราเตอร์ที่มีโปโตคอลแบบสเตติกอย่างเดียว

ประเภทโปรโตคอลในการเลือกเส้นทางแบบไดนามิก

ประเภทโปรโตคอลในการเลือกเส้นทางแบบไดนามิกยังแบ่งออกได้หลายรูปแบบ ดังเช่น โปรโตคอลเกตเวร์ภายนอกและเกตเวร์ภายใน, โปรโตคอลดิสแทนซ์เวกเตอร์และลิงค์สเตท

โปรโตคอลเกตเวร์ภายนอกและเกตเวร์ภายใน

การจัดแบบนี้แบ่งชนิดของโปรโตคอลออกเป็นโปรโตคอลเกตแวร์ภายนอก (exterior gateway protocol) และโปรโตคอลเกตแวร์ภายใน (interior gateway protocol)
ในปัจจุบันโปรโตคอลที่นิยมใช้ระหว่างเครือข่ายคือ บีพีจี (BPG: Border gateway protocol) จนกระทั่งพัฒนามาเป็น บีพีจี-4 (BPG-4) สำหรับโปรโตคอลเกตเวร์ภายในเป็นโปรโตคอลที่

ออกแบบเพื่อใช้งานในระบบอัตโนมัตดังเช่น อาร์ไอพี (RIP : Routing information Protocol) และ โอเอสพีเอฟ (OSPF :Open Shortest Path First) เป็นต้น

หลักการทำงานของอาร์ไอพี

อาร์ไอพีเป็นโปรโตคอลแบบดิสแตนซ์เวกเตอร์ ขั้นตอนวิธีการพื้นฐานที่อาร์เอพีใช้คำนวณหาระยะทางคือขั้นตอนวิธีของ ฟอร์ด/ฟูลเกอร์สัน หรือเรียกในอีกชื่อหนึ่งคือ เบลเมน-ฟอร์ด เราเตอร์อาร์ไอพีจะเก็บตารางเส้นทางซึ่งประกอบด้วย แอดเดรสเครือข่าย, แอสเดรสเราเตอร์ถัดไปซึ่งเป็นเกตเวย์ยังเครือข่ายนั้น, เมตริกประจำเส้นทางซึ่งโดยปกติจะนับตามจำนวนเราเตอร์ระหว่างทาง

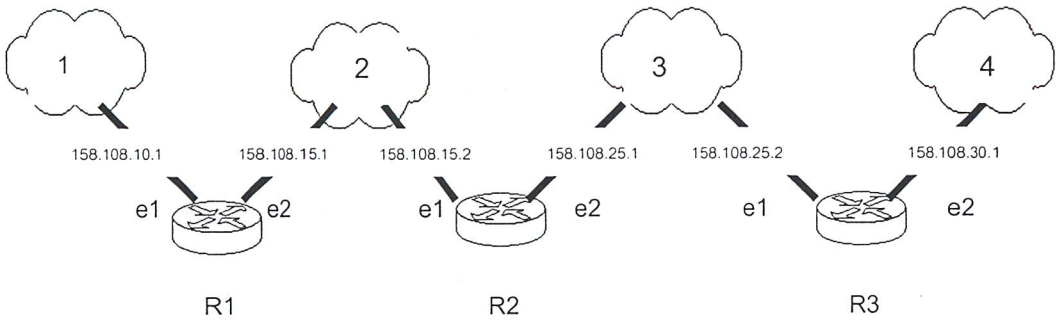
การทำงานของอาร์ไอพี ขั้นต้นเราเตอร์จะได้รับการกำหนดแอสเดรสเครือข่ายประจำแต่ละอินเทอร์เฟสของเราเตอร์เอง จากนั้นเราเตอร์จะสร้างตารางเส้นทางประจำตัวและประกาศตารางเส้นทางไปยังเราเตอร์ข้างเคียงด้วยการบอร์ดคาสต์ อาร์ไอพีบอร์ดคาสโดยใช้ยูดีพีผ่านพอร์ต 520

เมื่อเราเตอร์ได้รับตารางเลือกเส้นทางจากเราเตอร์อื่นๆ ก็จะทำการปรับเส้นทางประจำตัวและกระจายตารางออกไปอีก วิธีนี้ทำให้เราเตอร์แต่ละตัวสามารถคำนวณระยะทางและทิศทางของเครือข่ายอื่นที่เราเตอร์นั้นไม่ได้เชื่อมต่อโดยตรง และในที่สุดเราเตอร์ทุกตัวก็จะทราบถึงแอดเดรสของเครือข่ายทั้งหมด เราเตอร์อาร์ไอพีจะปรับค่าในตารางเส้นทางเมื่อได้รับการประกาศค่าจากเราเตอร์ข้างเคียงภายใต้เงื่อนไขดังต่อไปนี้ คือ

1. หากพบว่าเป็นเส้นทางใหม่ซึ่งไม่อยู่ในตาราง ให้ใส่เส้นทางนั้นเข้าสู่ตาราง
2. หากเป็นเส้นทางเก่าที่มีในตารางแต่ค่าที่ได้รับมีระยะทางที่สั้นกว่าให้แทนเส้นทางเก่าด้วยเส้นทางที่ได้รับมาใหม่
3. หากได้รับเส้นทางใดๆ จากเราเตอร์ R ใดๆ และตรวจพบในตารางว่ามีเส้นทางซึ่งเราเตอร์ R เป็นเกตเวย์อยู่แล้ว ให้ปรับค่าเส้นทางใหม่ตามค่าที่ได้รับล่าสุดจากเราเตอร์ R

ตัวอย่างการสร้างตารางเส้นทาง จากรูปนี้มีเราเตอร์อยู่ 3 ตัวเชื่อมต่อกันดังรูป โดยกำหนดไอพีแอสเดรสประจำแต่ละอินเทอร์เฟสไว้แล้ว กำหนดซับเน็ทมาร์กเป็น 255.255.255.0

แอดเดรสเครือข่าย 1 ถึง 4 เป็น 158.108.10.0, 158.108.15.0, 158.108.25.0 และ 158.108.30.0



รูปที่ 2.5 แสดงตัวอย่างการทำงานของอาร์ไอพี

เมื่อเราเตอร์ทั้งสามเริ่มต้นทำงาน ค่าในตารางเลือกเส้นทางของแต่ละเราเตอร์ซึ่งเกิดจากการติดตั้งค่าจากผู้ดูแลระบบเครือข่ายจะประกอบด้วยเครือข่ายที่เชื่อมอยู่โดยตรง ดังนี้

ตารางที่ 2.3 ตารางเส้นทางของเราเตอร์เมื่อเริ่มต้นการทำงาน

ตาราง R1

เครือข่าย	เกตเวย์	อินเทอร์เฟส	เมตริก
158.108.10.0	0.0.0.0	e1	1
158.108.15.0	0.0.0.0	e2	1

ตาราง R2

เครือข่าย	เกตเวย์	อินเทอร์เฟส	เมตริก
158.108.15.0	0.0.0.0	e1	1
158.108.25.0	0.0.0.0	e2	1

ตาราง R3

เครือข่าย	เกตเวย์	อินเทอร์เฟส	เมตริก
158.108.25.0	0.0.0.0	e1	1
158.108.30.0	0.0.0.0	e2	1

ตารางเลือกเส้นทางตามตัวอย่างข้างต้นนั้นประกอบด้วย

เครือข่าย : ไอพีแอสเดรสของเครือข่ายปลายทาง

เกตเวย์ : ไอพีแอสเดรสของเราเตอร์ซึ่งเป็นทางออกไปสู่เครือข่ายปลายทาง

อินเทอร์เฟส : อินเทอร์เฟสของเราเตอร์

เมตริก : จำนวนขั้น

เริ่มต้นนั้นเราเตอร์จะทราบเพียงเส้นทางไปยังเครือข่ายที่ติดกับเราเตอร์เท่านั้นส่วนเส้นทางไปยังเครือข่ายอื่นๆ จะสร้างจากการแลกเปลี่ยนตารางไปยังเราเตอร์ข้างเคียงทุกช่วงเวลาที่กำหนดค่าที่ใช้โดยปกติคือทุกๆ 30 วินาที หากสมมติให้ R1 ได้บอร์ดคลาสเป็นอันดับแรก ดังนั้น R2 จะได้รับแอดเดรส 158.108.10.0 และ 158.108.15.0 จาก R1

เครือข่าย 158.108.15.0 มีอยู่ในตารางแล้วและเป็นเครือข่ายที่ติดกับ R2 จึงไม่ต้องปรับค่า ส่วนเครือข่าย 158.108.10.0 เป็นค่าใหม่ ดังตารางดังต่อไปนี้

ตารางที่ 2.4 แสดงตารางเส้นทางของเราเตอร์

ตาราง R2

เครือข่าย	เกตเวย์	อินเทอร์เฟส	เมตริก
158.108.10.0	158.108.15.1	e1	2
158.108.15.0	0.0.0.0	e2	1
158.108.25.0	0.0.0.0	e2	1

จากนั้นให้ R3 ทำการบอร์ดคลาส พิจารณาเฉพาะเราเตอร์ R2 จะได้รับค่าและปรับตารางใหม่ได้เป็น

ตาราง R2

เครือข่าย	เกตเวย์	อินเทอร์เฟส	เมตริก
158.108.10.0	158.108.15.1	e1	2
158.108.15.0	0.0.0.0	e2	1
158.108.25.0	0.0.0.0	e2	1
158.108.30.0	158.108.25.2	e2	2

ตารางของ R2 นั้นจะมีค่าเส้นทางทั้งเครือข่ายแล้ว ซึ่งหากว่า R1 และ R3 ได้รับการบอร์ดคลาสจาก R2 ก็จะทราบถึงเครือข่ายอื่นๆ ทั้งหมดเช่นกัน

หลักการทำงานของโอเอสพีเอฟ

การที่เราเตอร์ใช้โปรโตคอลเลือกเส้นทางเพื่อสร้างแผนที่เครือข่าย โปรโตคอลลิงค์สเทตประเภทโอเอสพีเอฟทำหน้าที่สร้างแผนที่ให้กับเราเตอร์ 3 ขั้นตอน คือ ขั้นตอนแรกเราเตอร์จะตรวจหาเราเตอร์อื่นที่อยู่ข้างเคียง ขั้นที่สอง เราเตอร์จะประกาศค่านี้ออกไปให้เราเตอร์อื่นๆ ได้ทราบ ขั้นตอนสุดท้ายเราเตอร์จะอาศัยข้อมูลจากขั้นตอนที่สองสร้างแผนที่ไปยังเราเตอร์อื่นทุกตัวในเครือข่าย ซึ่งเราเตอร์ทุกตัวไขในเครือข่ายนั้นจะมีแผนที่ประจำตัวที่ทราบถึงเส้นทางทั้งหมดในเครือข่าย

ขั้นตอนแรก เมื่อเราเตอร์ในเครือข่ายที่ใช้โอเอสพีเอฟจะทำการเรียนรู้เครือข่ายโดย จะส่งแพ็กเก็ต “ทักทาย” หรือ ฮัลโหลแพ็กเก็ต (hello packet) ออกไปทุกๆ ลิงค์ของเราเตอร์ เช่น เราเตอร์ A จะส่งแพ็กเก็ตไปทักทายเราเตอร์ที่อยู่ B ,C และเราเตอร์ที่อยู่ที่ D ซึ่งในขณะเดียวกันเราเตอร์ ทั้งสามก็ส่งแพ็กเก็ตทักทายไปยังเราเตอร์ที่อยู่ที่ A ด้วย

ขั้นตอนที่สอง หลังจากที่ส่งแพ็กเกตไปแล้วเราเตอร์แต่ละตัวจะทำการสร้างฐานข้อมูลที่เก็บเส้นทางไปยังเราเตอร์ทุกตัวในเครือข่าย วิธีการที่ใช้คือเราเตอร์จะสร้างแพ็กเก็ตบรรจุเส้นทางที่มีอยู่ทั้งหมดในขณะนั้นและ “ปล่อย” (flooding) ให้แพ็กเก็ตไหลจากเราเตอร์หนึ่งไปอีกเราเตอร์หนึ่ง

เมื่อเราเตอร์ได้รับแพกเก็ตก็จะทำสำเนาแพ็กเก็ตนั้นส่งออกไปยังลิงค์อื่นด้วย เราเตอร์ต้องตรวจสอบว่าเป็นแพ็กเก็ตเก่าซ้ำของเดิมหรือเป็นแพ็กเก็ตใหม่ ซึ่งหากได้รับแพ็กเก็ตเก่าจากเราเตอร์อื่นให้กำหนดกำจัดทิ้งไป หากเป็นแพ็กเก็ตใหม่จะต้องปล่อยแพ็กเก็ตนั้นไปทุกอินเตอร์เฟสยกเว้นอินเตอร์เฟสที่ได้รับแพ็กเก็ต

รูปแบบแพ็กเก็ตที่เราเตอร์ส่งออกไปนั้นเรียกว่า “แพ็กเก็ตประกาศลิงค์สเทต” (link state advertisement) หรือ แพ็กเก็ตแอลลเอสเอ (LSA) เมื่อเราเตอร์ปล่อยแพ็กเก็ตนี้ เพียงชั่วระยะเวลาหนึ่งแพ็กเก็ตจะกระจายไปทั่วทั้งเครือข่าย ซึ่งท้ายที่สุดเราเตอร์ทุกตัวจะทราบเส้นทางไปยังเราเตอร์ตัวอื่นๆ และได้ตารางจากการหาเส้นทางนี้เรียกว่า ฐานข้อมูลลิงค์สเทต (link state database)

ขั้นตอนสุดท้าย

ฐานข้อมูลที่ได้จากขั้นตอนที่แล้วยังไม่ใช่ตารางเส้นทางที่นำไปใช้ได้ ตารางเส้นทางจะได้จากการคำนวณหาเส้นทางที่สั้นที่สุดไปยังเราเตอร์อื่น ซึ่งวิธีที่โปรโตคอลลิงค์สเทตใช้คือ การหาเส้นทางที่สั้นที่สุดตามขั้นตอนของไดจ์สตรา (Dijkstra algorithm)

ขั้นตอนตามแบบของไดจ์สตรา เราเตอร์จะสร้างเส้นทางที่เขียนแทนด้วยต้นไม้กำกับทิศทางโดยมีกิ่งเชื่อมไปยังเราเตอร์อื่น รากของต้นไม้ คือเราเตอร์ที่เป็นจุดตั้งต้นคำนวณหาเส้นทาง แต่ละเส้นจะกำกับด้วยเลเบลซึ่งแบ่งเส้นทางออกเป็นสองชนิดคือ เส้นทางชั่วคราว และเส้นทางถาวร ซึ่งเส้นทางชั่วคราวใช้กำกับเส้นที่ค้นพบแต่ยังไม่ผ่านการคำนวณว่าเป็นเส้นทางที่สั้นที่สุด ส่วนเส้นทางถาวรนั้น เป็นเส้นทางที่ได้ทำการคำนวณแล้วว่าสั้นที่สุด

เมื่อเริ่มต้นทำงานที่เราเตอร์ใด เราเตอร์นั้นจะมีระยะทางถึงตัวเองมีค่าเป็นศูนย์และสร้างเส้นทางชั่วคราวไปยังเราเตอร์ที่อยู่ข้างเคียงก่อน ส่วนเส้นทางที่ไปยังเราเตอร์อื่นนั้นให้ทีค่าเป็นอนันต์

2.4 โปรโตคอลไอพี

ไอพีนับเป็นโปรโตคอลแกนของทีซีพี/ไอพี โปรโตคอลอื่นไม่ว่าจะเป็นทีซีพีหรือยูดีพีต้องจัดข้อมูลในรูป เดทาแกรม (datagram) ซึ่งประกอบด้วยเฮดเดอร์และข้อมูลตามรูปที่ 2.6

หน้าที่หลักของไอพีคือจัดขนาดของข้อมูลให้พอเหมาะและเลือกเส้นทางที่เหมาะสมเพื่อจัดส่งเดทาแกรม ไอพีมีรูปแบบการจัดส่งเดทาแกรมเป็นแบบ “unreliable” และ “connectionless”

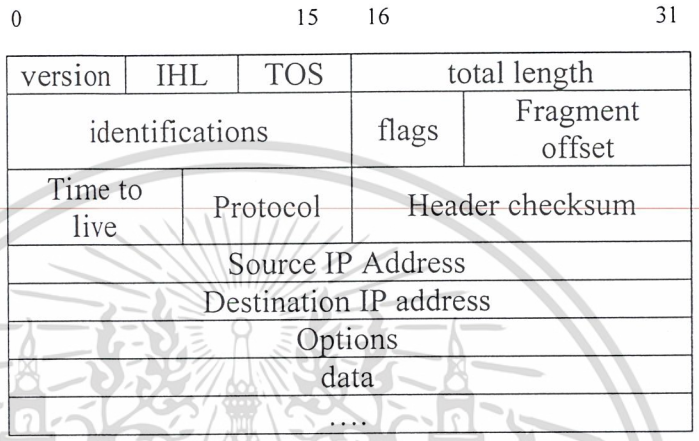
รูปที่ 2.6 ฟอร์แมตของเดทาแกรม

ความหมายของ “unreliable” คือไอพีไม่มีกลไกรับประกันว่าเดทาแกรมที่ส่งจะไปถึงปลายทางได้สำเร็จ ไอพีให้บริการลำเลียงเดทาแกรมอย่างดีที่สุด หากมีความผิดปกติใดเกิดขึ้นระหว่างการนำส่งเดทาแกรม เช่นบัฟเฟอร์ของเราเตอร์ระหว่างทางจนเต็มไม่สามารถรับเดทาแกรมได้ สิ่งที่ไอพีดำเนินการกับเดทาแกรมนั้นไป แล้วรายงานสาเหตุของปัญหากลับไปด้วยโปรโตคอลไอซีเอ็มพี

ความหมายของ “connectionless” คือไอพีไม่สถาปนาการเชื่อมโยงเพื่อกำหนดเส้นทางลำเลียงระหว่างต้นทางและปลายทาง ไอพีไม่เก็บสถานะใดๆของเดทาแกรมที่ส่งออกไปเดทาแกรมแต่ละชิ้นจึงเป็นอิสระจากกัน และมีโอกาสไปถึงปลายทางโดยไม่เรียงลำดับ เช่น สถานีต้นทางส่งเดทาแกรม A และ B ตามลำดับ ทั้ง A และ B อาจใช้เส้นทางต่างกันทำให้ B อาจไปถึงปลายทางก่อน A ได้

2.4.1 ไอพีเดทาแกรม

ไอพีเดทาแกรมมีฟอร์แมตรูปที่ 2.7 และเก็บตัวเลขฐานสองตามแบบ บิ๊กเอ็นเดียน (big endian) เฟรมฟอร์แมตในทีซีพี/ไอพี นิยมเขียนแสดงเป็นแถวๆ ละ 32 บิต เรียงจากซ้ายไปขวาและจากบนลงล่าง ซีพียูบางรุ่นที่เก็บตัวเลขฐานสองตามแบบ ลิตเติ้ลเอ็นเดียน (little endian) จะต้องแปลงลำดับให้ถูกต้องก่อนที่จะส่งข้อมูล



รูปที่ 2.7 ไอพีเดทาแกรม

- version ขนาด 4 บิต : แสดงโปรโตคอล รุ่นที่ใช้งานขณะปัจจุบันมีค่า 4
- Internet Header Length (IHL) ขนาด 4 บิต : บอกความยาวเฉพาะเฮดเดอร์ของเดทา แกรมโดยนับจาก version จนถึงไบต์สุดท้ายก่อนที่จะถึงข้อมูล หน่วยนับความยาวจะบอกเป็นจำนวนเท่าของ 4 ไบต์ (หรือ 32 บิตเวิร์ด) หาก IHL มีค่าเท่ากับ5 จะหมายถึงส่วนหัวมีขนาด 20 ไบต์ซึ่งเป็นค่าที่บอกว่าไม่มี option และ padding อยู่ในเดทาแกรม
- Type of Service (TOS) ขนาด 8 บิต : ฟิลด์นี้ใช้กำหนดรูปแบบการให้บริการตามลักษณะโปรโตคอลแอพลิเคชัน
- total length มีขนาด 16 บิต : บอกถึงความยาวทั้งหมดของเดทาแกรม (เฮดเตอร์และข้อมูล) โดยมีหน่วยนับเป็นไบต์ เนื่องจากฟิลด์นี้มีขนาด 16 บิต ไอพีเดทาแกรมจึงมีขนาดใหญ่สุดเท่ากับ 2^{16}-1 หรือ 65,535 ไบต์
- identification ขนาด 16 บิต
- flags ขนาด 3 บิต
- fragment offset ขนาด 13 บิต

- Time to Live (TTL) ขนาด 8 บิต : ฟิลด์นี้ใช้กำหนดจำนวนเราเตอร์ที่เดทาแกรมจะเดินทางผ่านได้หรืออีกนัยหนึ่งคือกำหนดอายุของเดทาแกรมซึ่งมีค่าได้สูงสุดตามขนาดฟิลด์คือ 2^8-1 หรือ 255 สถานีที่ส่งเดทาแกรมจะตั้งค่า TTL ไว้ที่ค่าใดค่าหนึ่ง เราเตอร์ที่รับเดทา แกรมจะปรับลดค่านี้ลงหนึ่งหน่วย หากลดลงเป็น 0 เราเตอร์จะทิ้งเดทาแกรมนั้นและรายงานกลับไปด้วยไอซีเอ็มพี วิธีนี้ช่วยป้องกันปัญหาเดทาแกรมวนรอบ (routing loop) สถานีต้นทางต้องเลือกใช้ค่านี้ให้เหมาะสม เนื่องจากหากมีค่าน้อยไปจะทำให้เดทาแกรมเดินทางไปไม่ถึงปลายทาง หรือหากตั้งไว้มากเกินไปก็จะสร้างภาระให้ระบบเมื่อมีความผิดปกติด้านการเลือกเส้นทาง ค่าโดยปกติที่ใช้คือ 64
- Protocol ขนาด 8 บิต : ฟิลด์บอกชนิดของโปรโตคอลระดับบนที่เอ็นแคปซูเลต ในเดทาแกรม เพื่อให้สถานีปลายทางและสามารถส่งข้อมูลไปยังโปรโตคอลระดับบนได้ถูกต้อง ค่าที่ใช้ประจำโปรโตคอลมีดังเช่น

1	ICMP
6	TCP
17	UDP
89	OSPF

- header checksum ขนาด 16 บิต : ใช้ตรวจสอบความผิดพลาดเฉพาะเฮดเดอร์โดยไม่รวมส่วนข้อมูล การคำนวณผลรวมตรวจสอบจะเริ่มต้นด้วยการให้ฟิลด์ checksum มีค่าเป็น 0 จากนั้นจึงบวกเฮดเดอร์ครั้งละ 16 บิตแบบเติมเต็มหนึ่ง (1's complement) เมื่อได้ผลลัพธ์แล้ว จะนำใส่ในฟิลด์ checksum ไอพีปลายทางเมื่อได้รับเดทาแกรมแล้วก็เพียงแต่บวก เฮดเดอร์ทั้งหมดครั้งละ 16 บิตแบบเติมเต็มหนึ่ง
- source IP address ขนาด 32 บิต : กำหนดไอพีแอดเดรสต้นทาง
- destination IP address ขนาด 32 บิต : กำหนดไอพีแอสเดรสปลายทาง
- option ขนาดไม่คงที่ : ใช้สำหรับกำหนดข่าวสารเพิ่มเติมสำหรับเดทาแกรม ค่าที่ใช้ในปัจจุบันจะเกี่ยวข้องกับการรักษาความปลอดภัย และการบันทึกผลลัพธ์จากการทำงานของคำสั่ง trace route หรือ ping
- padding ขนาด 0 ถึง 3 ไบต์ : ใช้สำหรับผนวกเพิ่มเพื่อให้จำนวนไบต์ของ option รวมกับ padding เป็นจำนวนเท่าของ 32 บิต ค่าในฟิลด์ padding จึงไม่มีความสำคัญใด
- data ขนาดไม่คงที่ : ข้อมูลจากโปรโตคอลระดับบน

2.5 โปรโตคอลทีซีพี

ทีซีพีเป็นโปรโตคอลที่ให้บริการชนิดที่ต้องมีการเชื่อมต่อ รับประกันความเชื่อถือ ในการจัดลำเลียงข้อมูล ทีซีพีรับประกันความเชื่อถือโดยทำหน้าที่ตรวจสอบเซกเมนต์ที่ผิดปกติและจัดส่งเซกเมนต์ซ้ำใหม่ รวมทั้งจัดลำดับให้ถูกต้องก่อนส่งไปยังโปรแกรมประยุกต์ระดับบนเฮดเดอร์และข้อมูลทีซีพีเรียกว่า เซกเมนต์ (segment) การซ่อนรายละเอียดทีซีพีเซกเมนต์ในไอพีเดทาแกรมแสดงได้รูปที่ 2.8

รูปที่ 2.8 การซ่อนรายละเอียดทีซีพี

สถานีต้นทางจะต้องสถาปนาการเชื่อมโยงกับสถานีปลายทางก่อนที่ทีซีพีเซกเมนต์ การสถาปนาการเชื่อมโยงใช้ประโยชน์เพื่อให้มั่นใจว่าปลายทางพร้อมที่จะสื่อสารด้วย และเมื่อเสร็จสิ้นการส่งถ่ายข้อมูลแล้วก็จะปิดการเชื่อมโยง

การสถาปนาการเชื่อมโยงของทีซีพีอาจเปรียบเทียบได้กับการติดต่อทางโทรศัพท์กล่าวคือ เมื่อหมุนเลขปลายทางแล้ว ผู้เรียกต้องรอให้ปลายทางรับสาย เมื่อทักทายและแจ้งให้ทราบว่าใครเป็นผู้เรียกสายแล้วจึงเริ่มการสนทนา เนื่องจากทีซีพีทำงานตามแบบไคลเอ็นต์-เซิร์ฟเวอร์ ไคลเอ็นต์จะเป็นผู้ร้องขอบริการและขอสถาปนาการเชื่อมโยง ส่วนเซิร์ฟเวอร์รับการร้องขอและให้บริการต่อไคลเอ็นต์ การขนถ่ายข้อมูลระหว่างไคลเอ็นต์และเซิร์ฟเวอร์เป็นแบบฟูลดูเพล็กซ์เสมือนท่อลำเลียงสองท่อต่อเชื่อมระหว่างไคลเอ็นต์และเซิร์ฟเวอร์

2.5.1 บริการรับประกันความเชื่อถือของทีซีพี

ทีซีพีให้บริการจัดการด้านความเชื่อถือการลำเลียงเซกเมนต์ที่สำคัญคือ การตรวจจับและแก้ไขข้อผิดพลาด การควบคุมปริมาณการไหลข้อมูล การจัดลำดับ และการกำจัดเซกเมนต์ซ้ำ โดยมีรายละเอียดดังต่อไปนี้

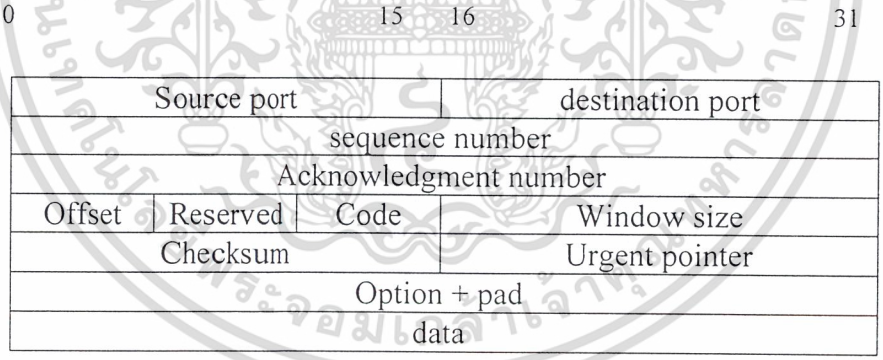
1. การตรวจจับและแก้ไขข้อผิดพลาดมุ่งเน้นการแก้ปัญหาเซกเมนต์ที่ผิดปกติจากปัญหาของสายสื่อสารหรือโปรโตคอลระดับล่าง หรือเซกเมนต์เดินทางไปไม่ถึง

ปลายทาง ทีซีพีจัดการกับปัญหานี้โดยใช้ผลรวมตรวจสอบเช่นเดียวกับยูดีพี หารค่าผลรวมตรวจสอบไม่ถูกต้อง ทีซีพีจะส่งเซกเมนต์ซ้ำใหม่

2. เมื่อทีซีพีได้รับเซกเมนต์จะจะตอบรับกลับไปต้นทาง ทีซีพีใช้การตอบรับเพื่อยืนยันว่าได้รับข้อมูลอย่างถูกต้อง
3. ทีซีพีจะตั้งเวลาเมื่อส่งเซกเมนต์เพื่อรอการตอบรับจากปลายทาง หากไม่มีการตอบรับภายในเวลาที่กำหนดทีซีพีจะส่งเซกเมนต์ซ้ำ วิธีนี้เรียกว่า การตอบรับแบบบวกพร้อมกับการส่งซ้ำ (positive acknowledges with retransmission)
4. ทีซีพีฝ่ายรับสามารถกำหนดให้ฝ่ายส่งจัดส่งข้อมูลเป็นปริมาณเท่าที่จะรับได้จริงตามขนาดบัฟเฟอร์ที่มีอยู่ การควบคุมปริมาณการไหลข้อมูลเป็นการป้องกันไม่ให้เซกเมนต์ไปเกินขีดความสามารถของฝ่ายรับ
5. หากเซกเมนต์มาถึงไม่เป็นลำดับ ทีซีพีฝ่ายรับต้องจัดลำดับเซกเมนต์ให้ถูกต้อง
6. ทีซีพีมีหน้าที่จัดเซกเมนต์ซ้ำซ้อน

2.5.1 ทีซีพีเฮดเดอร์

ทีซีพีเฮดเดอร์ประกอบด้วยฟิลด์จำนวนมากที่ให้บริการตามฟังก์ชันที่กล่าวข้างต้น รูปที่ 2.9 แสดงเฮดเดอร์ของทีซีพี แต่ละฟิลด์มีความหมายดังต่อไปนี้



รูปที่ 2.9 ทีซีพีเฮดเดอร์

- source port ขนาด 16 บิต : หมายเลขของสถานีต้นทาง
- destination port ขนาด 16 บิต : หมายเลขพอร์ตของสถานีปลายทาง
- sequence number ขนาด 32 บิต : ทีซีพีใช้ เลขลำดับ เป็นตัวชี้ตำแหน่งข้อมูลไบต์แรกที่จะจัดส่ง หมายเลขเริ่มต้นไม่จำเป็นต้องเริ่มด้วย 1 แต่อาจเริ่มด้วยค่าใดๆ ก็ได้ ข้อมูลในเซกเมนต์ถัดไปจะมีเลขลำดับที่สัมพันธ์เลขลำดับในเซกเมนต์ก่อนหน้า

- acknowledgement number ขนาด 32 บิต : ค่ากำหนด เลขตอบรับซึ่งใช้ตอบกลับไปว่าได้รับข้อมูลแล้ว เลขตอบรับจะมีค่าเท่ากับเลขลำดับประจำเซกเมนต์บวกด้วยจำนวนไบต์ข้อมูลและบวกด้วยหนึ่ง เช่น เซกเมนต์หนึ่งมีเลขลำดับเท่ากับ 21 และมีข้อมูลตั้งแต่ต้นถึงไบต์ลำดับที่ 41 แล้ว และคาดว่าไบต์ถัดไปคือ ไบต์ที่ 42
- offset (data offset) ขนาด 4 บิต : บอกถึงตำแหน่งเริ่มต้นของไบต์ข้อมูลหรืออีกนัยหนึ่งใช้บอกขนาดเฮดเดอร์ ตังเลขนี้มีหน่วยจำนวนเท่าของ 4 ไบต์เช่นเดียวกับที่ใช้ในไอพี เดทาแกรม เฮดเดอร์ของทีซีพีมีความยาวขึ้นกับฟิลด์ออฟชัน ตัวเลขในฟิลด์ offset จะเท่ากับ 5 ซึ่งเท่ากับ 20 ไบต์ (5*4= 20) สำรองไว้ในอนาคต
- code ประกอบด้วย 6 ฟิลด์ย่อย แต่ละฟิลด์ย่อยมีขนาด 1 บิต ทำหน้าที่เป็นแฟล็กเรียงลำดับจากซ้ายไปขวา เรียงลำดับดังนี้

  - URGent ถ้าบิตนี้เป็น " 1 " หมายความว่า Urgent pointer บรรจุข้อมูลที่ต้องรีบดำเนินการเร่งด่วน
  - ACKnowledgement ถ้าบิตนี้เป็น " 1 " หมายถึงเป็นเซกเมนต์ตอบรับ โดยตอบอ้างอิงเลขตามที่กำหนดในฟิลด์ Acknowledgement number
  - PuSh ถ้าบิตนี้เป็น " 1 " หมายความว่าทันทีที่สถานีปลายทางได้รับเซกเมนต์ต้องรีบส่งข้อมูลไปยังโปรโตคอลประยุกต์ทันทีโดยไม่ต้องรอให้บัฟเฟอร์เต็ม
  - ReSeT ถ้าบิตนี้เป็น " 1 " หมายถึงให้ยกเลิกการเชื่อมต่อนี้ เนื่องจากอาจมีความผิดปกติเกิดขึ้นระหว่างคู่สถานีที่กำลังติดต่อกันอยู่ หากจำเป็นต้องส่งข้อมูลระหว่างกันอีกก็ต้องเริ่มต้นสถาปนาการเชื่อมต่อใหม่
  - SYNchronize ถ้าบิตนี้เป็น " 1 " หมายถึงขอเริ่มต้นสถาปนาการเชื่อมต่อและเมื่อการสถาปนาเสร็จสิ้น บิตนี้จะถูกกำหนดให้เป็น " 0 " หลังจากนั้นจึงสามารถส่งผ่านข้อมูลระหว่างกันได้
  - FINish ถ้าบิตนี้เป็น " 1 " หมายถึงขอจบการเชื่อมต่อ

window size ขนาด 16 บิต : สถานีปลายทางใช้ฟิลด์นี้แจ้งขนาดบัฟเฟอร์ที่มีอยู่ (หน่วยเป็นไบต์) สถานีที่ติดต่อด้วยต้องไม่ส่งข้อมูลเกินค่านี้

checksum ขนาด16 บิต : ผลรวมตรวจสอบความถูกต้องของเซกเมนต์โดยคำนวณทั้งเฮดเดอร์และข้อมูล (ใช้เฮดเดอร์เทียบเช่นเดียวกับยูดีพี)

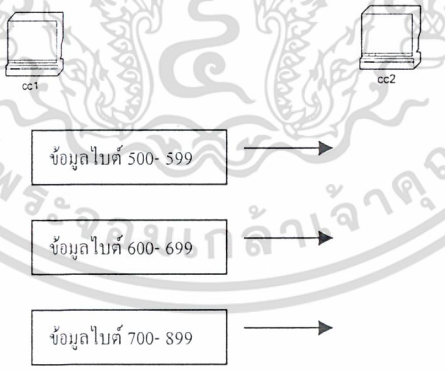
urgent pointer ขนาด 16 : พอยเตอร์ชี้ตำแหน่งไบต์ข้อมูลที่ต้องดำเนินการเร่งด่วนที่ต้องการให้โปรแกรมประยุกต์ดำเนินการทันที ค่าที่บรรจุในฟิลด์นี้จะมีความหมายก็ต่อเมื่อแฟล็ก URG ถูกเซ็ตเป็น " 1 "

options ขนาดแปรเปลี่ยนได้ : ใช้กำหนดงานเพิ่มเติมให้กับทีซีพีซึ่งจะมีหรือไม่มีก็ได้หากฟิลด์ offset หากมีค่าเป็น 5 แสดงว่ามีเฮดเดอร์มีขนาด 20 ไบต์ซึ่งหมายถึงไม่ใช้ออฟชัน

pad ขนาด 0 ถึง 24 บิต : ใช้เป็นส่วนที่ทำให้ขนาดของออฟชันเป็นจำนวนเท่าของ 32 บิต เพื่อให้เฮดเดอร์ลงตัวที่ค่าจำนวนเท่าของ 32

2.5.3 การถ่ายโอนโดยใช้เลขลำดับ

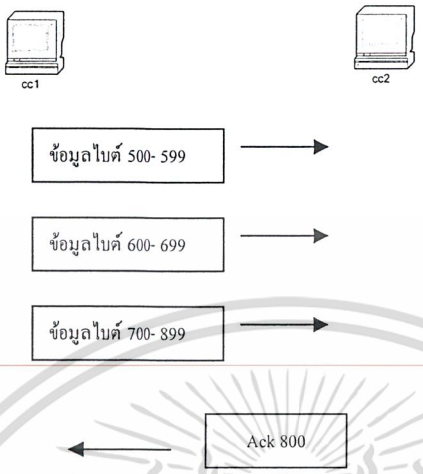
ทีซีพีอาศัยเลขลำดับและเลขตอบรับเพื่อยืนยันการถ่ายโอน ตามที่กล่าวมาแล้วเลขลำดับที่ใช้ในเฮดเดอร์ไม่ใช่เลขลำดับเซกเมนต์ หากแต่เป็นเลขชี้ตำแหน่งไบต์ข้อมูลที่กำลังจัดส่ง ในขั้นตอนแรกของการสถาปนาการเชื่อมต่อแต่ละครั้งทีซีพีด้านไคลเอ็นต์จะเลือกเลขลำดับเริ่มต้นขนาด 32 บิต แล้วส่งเซกเมนต์ที่กำกับด้วยเลขลำดับเริ่มต้นเพื่อขอสถาปนาการเชื่อมต่อ เลขลำดับนี้จะเพิ่มค่าไปตามจำนวนไบต์ที่ส่งในแต่ละครั้งตามรูปที่ 2.10 ซึ่งแสดงเลขลำดับเริ่มต้นที่ 500 และเซกเมนต์ถัดไปกำกับด้วยเลขลำดับ 600 และ 700 (ทีซีพี ไม่ได้ส่งตัวเลขกำกับขนาดไปในเซกเมนต์โดยตรงเหมือนในรูปนี้ ตัวเลข 599,699 และ 799 เป็นเพียงตัวเลขที่แสดงสมมติให้ทราบถึงจำนวนไบต์ข้อมูลที่ส่งในแต่ละเซกเมนต์)



รูปที่ 2.10 การส่งโดยใช้เลขลำดับ

ทีซีพีที่ได้รับเซกเมนต์จะตอบกลับด้วยตัวเลขตอบรับดังตัวอย่างในรูปที่ 2.11 สถานี cc2 ส่งเซกเมนต์กลับมาด้วยค่าเลขตอบรับ 800 ซึ่งหมายถึงได้รับข้อมูลตั้งแต่ต้นถึงเลขลำดับ 799 แล้ว

และคาดว่าจะรอรับไบต์ลำดับต่อไป หากไม่มีการตอบรับภายในเวลาที่กำหนดก็จะนำส่งเซกเมนต์นั้นซ้ำใหม่



รูปที่ 2.11 การตอบรับโดยใช้เลข

เลขลำดับมีขนาด 32 บิต ค่าสูงสุดที่เป็นไปได้คือ $2^{32} - 1$ เมื่อเลขลำดับเพิ่มถึงค่าสูงสุดแล้วจะวนกลับมาที่ 0 ใหม่อีกครั้งหนึ่ง แต่ในเครือข่ายความเร็วสูงที่ใช้ในปัจจุบันถึงแม้มีการถ่ายโอนข้อมูลอย่างต่อเนื่องก็จะไม่พบปัญหาเลขลำดับวนกลับมาซ้ำค่าเดิมในระยะเวลาสั้นๆ ด้วยเหตุนี้หากมีเซกเมนต์ที่มีเลขลำดับซ้ำกับที่เคยได้รับไปแล้ว ทีซีพีถือว่าเกิดจากปัญหาการส่งเซกเมนต์ซ้ำ ซึ่งฝ่ายรับจะกำจัดเซกเมนต์ทิ้งไป

2.5.4 กลไกการทำงานของทีซีพี

ทีซีพีมีกลไกการลำเลียงข้อมูลระหว่างต้นทางและปลายทางหลายส่วน ในที่นี้จะกล่าวเฉพาะกลไกที่สำคัญ 3 อย่างก่อน ได้แก่ การสถาปนาการเชื่อมโยง การถ่ายโอนข้อมูล และการเลิกการเชื่อมโยงตามหัวข้อต่อไปนี้

2.5.4.1 การสถาปนาการเชื่อมโยง

เพื่อให้ผู้อ่านมองเห็นภาพขั้นตอนการทำงานของทีซีพี ขอยกตัวอย่างการสถาปนาการเชื่อมต่อระหว่างสถานีสองเครื่อง คือ cc1 และ cc2 โดยให้ cc1 เป็นฝ่ายขอบริการจาก cc2 ดังนั้นโปรโตคอลประยุกต์ด้าน cc1 จะเป็นไคลเอ็นต์ และโปรโตคอลประยุกต์ด้าน cc2 จะเป็นเซอร์ฟเวอร์

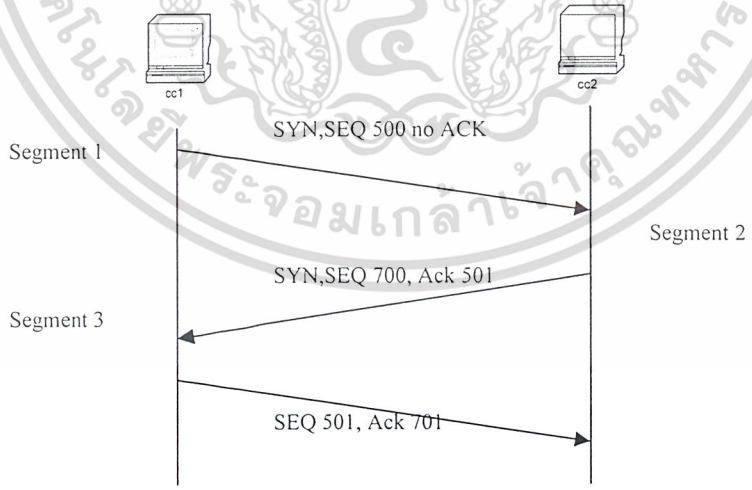
cc2 จะอยู่ในสภาพที่เรียกว่า การเปิดแบบพาสซีฟ (passive open) คือเซอร์ฟเวอร์สั่งให้ทีซีพีรอรับการเชื่อมต่อ ส่วน cc1 จะเริ่มการติดต่อเมื่อไคลเอ็นต์สั่งงาน การปิดการเชื่อมต่อจากไคลเอ็นต์เรียกว่า การเปิดแบบแอกทีฟ (active open) กระบวนการนี้ประกอบด้วยการส่งเซกเมนต์ 2 เซกเมนต์ดังรูปที่ 2.12 และมีลายระเอียดดังนี้

1. ทีซีพีของ cc1 เลือกเลขลำดับเริ่มต้น (ในที่นี้คือ 500) แล้วส่งเซกเมนต์ที่บรรจุเลขลำดับนี้ไปยัง cc2 พร้อมทั้งเซ็ตแฟล็ก SYN ให้เป็น " 1 " ขอให้สังเกตว่าเซกเมนต์นี้ (เซกเมนต์ 1) แฟล็ก ACK จะมีค่าเป็น " 0 "

2. ทีซีพีของ cc2 ได้รับเซกเมนต์จาก cc1 ก็จะเลือกเลขลำดับด้วยเซกเมนต์ SYN (เซกเมนต์2) พร้อมทั้งเซ็ตแฟล็ก ACK เพื่อแจ้งว่าได้รับเซกเมนต์ 1 โดยใช้เลขลำดับที่ได้รับจากทาง cc1 บวกด้วยหนึ่ง (ในที่นี้คือ 501)

3. ทีซีพีของ cc1 จะส่งเซกเมนต์ตอบรับกลับไป (เซกเมนต์ 3) โดยเซ็ตแฟล็ก ACK และใช้เลขลำดับที่ได้รับจาก cc2 บวกด้วยหนึ่ง

ต่อจากนั้นทีซีพีของ cc1 จะแจ้งไปยังโปรโตคอลระดับบนว่าเชื่อมต่อแล้วส่วนทีซีพีของ cc2 เมื่อได้รับเซกเมนต์ตอบรับ (เซกเมนต์ 3) ก็จะแจ้งขึ้นไปยังโปรโตคอลระดับบนว่าเชื่อมต่อแล้วเช่นกันเมื่อสิ้นสุดขั้นตอนนี้การสถาปนาทั้งสองด้านก็จะสมบูรณ์และพร้อมที่จะส่งข้อมูลได้ กระบวนการสถาปนาแบบนี้เรียกว่า three way handshake เพราะใช้ 3 เซกเมนต์โต้ตอบระหว่างกัน เซกเมนต์ทั้งสามนี้นิยมเรียกว่า SYN, SYN และ ACK ตามลำดับ

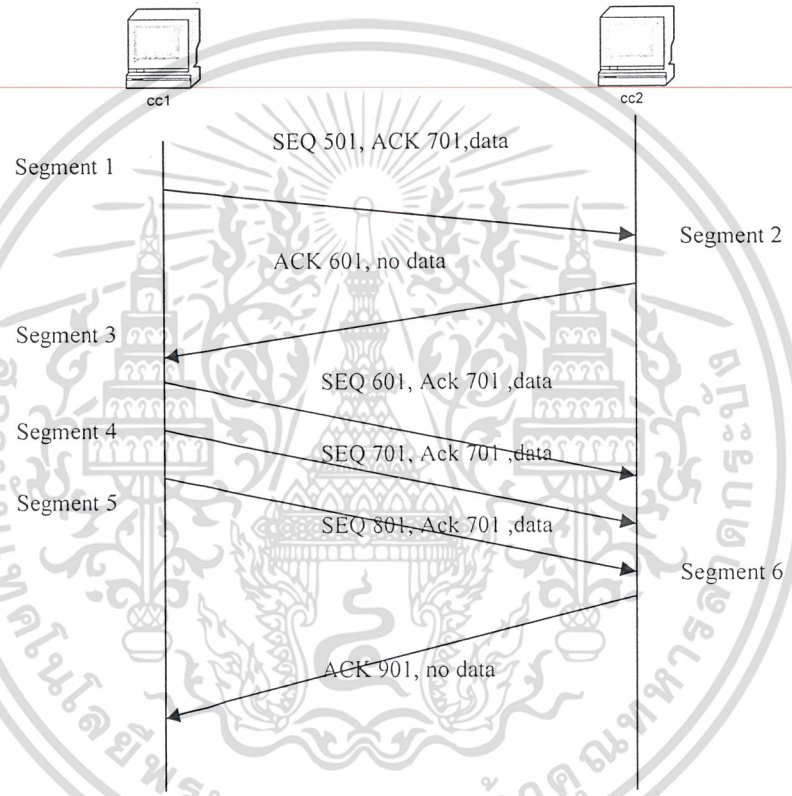


รูปที่ 2.12 การสถาปนาการเชื่อมโยงด้วยทีซีพี

2.5.4.2 การถ่ายโอนข้อมูล

การขนถ่ายข้อมูลเริ่มได้หลังจากการสถาปนาเสร็จสิ้นแล้ว รูปที่ 2.13 แสดงขั้นตอนโดยสมมติให้ cc1 ส่งข้อมูลไปยัง cc2 ครั้งละ 100 ไบต์ จำนวน 4 ครั้ง

เซกเมนต์แรกที่ cc1 ส่งไปยัง cc2 บรรจุข้อมูลไบต์ 501 ถึง 600 และใช้เลขตอบรับ 701 เมื่อ cc2 รับข้อมูลเซกเมนต์ 1 แล้ว ก็จะตอบกลับด้วยเซกเมนต์ 2 พร้อมตอบรับกำหนดข้อมูลที่คาดว่าจะได้รับต่อไปคือ 601



รูปที่ 2.13 ขั้นตอนการถ่ายโอนข้อมูล

cc1 สามารถส่งเซกเมนต์ไปอย่างต่อเนื่องได้ เช่นในแผนภาพมีการส่ง 3 เซกเมนต์คือ ไบต์ 601,701 และ 801 ทาง cc2 สามารถตอบกลับในคราวเดียวกันได้ (เซกเมนต์ 6)

ในกรณีที่มีการส่งข้อมูลทั้งสองทิศทางระหว่าง cc1 และ cc2 หลักการทั่วไปยังคงเป็นเช่นเดียวกัน เป็นเพียงการตอบรับพร้อมกับการส่งข้อมูลระหว่างกัน

2.5.4.3 การยกเลิกการเชื่อมต่อ

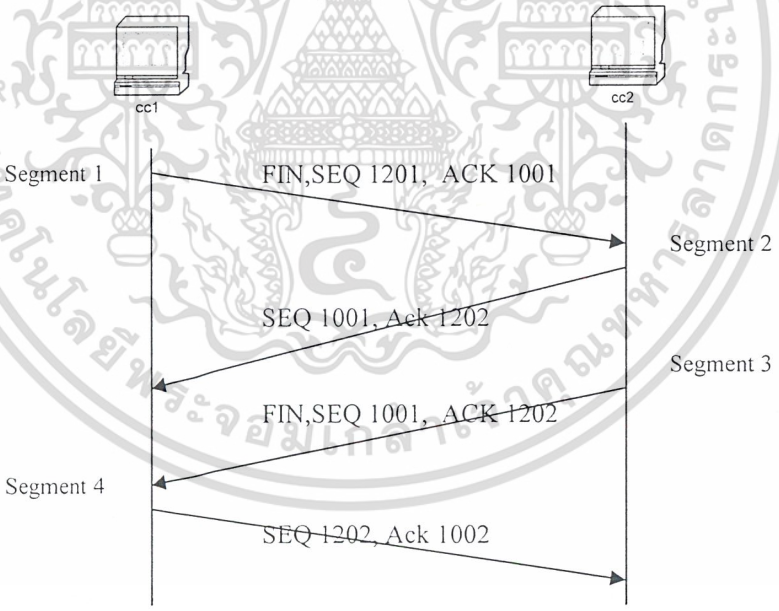
เนื่องจากการถ่ายโอนในทีซีพีเป็นแบบฟูลดูเพ็กซ์ การยกเลิกการเชื่อมต่อจึงต้องมีขั้นตอนเกิดขึ้นทั้งสองด้านดังรูปที่ 2.14 ในที่นี้สมมติว่า cc1 ไม่มีข้อมูลส่งอีกต่อไปจึงต้องเป็นฝ่ายขอการปิดการเชื่อมต่อ โปรโตคอลประยุกต์ของ cc1 จะแจ้งไปยังทีซีพีว่าส่งข้อมูลหมดแล้ว ถัดจากนั้นจะมีการแลกเปลี่ยนเซกเมนต์จำนวน 4 เซกเมนต์ดังนี้

1. ทีซีพีของ cc1 ส่งเซกเมนต์พร้อมที่มีเลขลำดับและเลขตอบรับตามปกติแต่เซ็ตแฟล็ก FIN เป็น " 1 "

2. เมื่อทีซีพีของ cc2 ได้รับเซกเมนต์ FIN จะส่งเซกเมนต์ตอบรับ (เซกเมนต์ 2) และแจ้งไปขึ้นยังโปรแกรมประยุกต์ว่า cc1 ขอปิดการเชื่อมต่อ โปรแกรมประยุกต์ของ cc2 จะแจ้งกลับมายังทีซีพีว่าปิดการเชื่อมต่อได้

3. ทีซีพี cc2 ส่งเซกเมนต์ FIN ไปยัง cc1 (เซกเมนต์ 3)

4. เมื่อ cc1 ได้รับเซกเมนต์ FIN จะตอบรับกลับไป (เซกเมนต์ 4) และแจ้งไปยังโปรแกรมประยุกต์ว่าการเชื่อมต่อปิดลงแล้ว



รูปที่ 2.14 การปิดการเชื่อมต่อ

2.5.4.4 การควบคุมกระแสข้อมูล

ทีซีพีอาศัยเลขลำดับเพื่อการจัดข้อมูล เลขลำดับจึงเป็นกุญแจสำคัญในการควบกระแสข้อมูลระหว่างฝ่ายรับและฝ่ายส่ง ไม่ให้ข้อมูลเกินกว่าที่จะรับได้ การควบคุมกระแสข้อมูล (flow control) ของทีซีพีทำงานภายใต้วิธี การเลื่อนหน้าต่าง (sliding windows)

เทคนิคการเลื่อนช่องหน้าต่างไม่ได้เป็นเทคนิคที่ใช้เฉพาะทีซีพีเท่านั้น หากแต่เป็นเทคนิคทั่วไปที่นำมาใช้ในที่โปรโตคอลระดับชั้นทรานสพอร์ตอื่นๆ ด้วย เพื่อให้ผู้อ่านทำความเข้าใจได้ทั้งกระบวนการจะทบทวนถึงหลักพื้นฐานและประโยชน์ของเทคนิคดังกล่าวก่อน

2.5.4.5 การเลื่อนหน้าต่าง

ลองพิจารณาถึงสภาวะการถ่ายโอนระหว่างไคลเอ็นต์ cc1 และ เซอร์ฟเวอร์ cc2 หาก cc1 ใช้หลักการตอบรับเป็นรายเซกเมนต์คือทุกครั้งที่ส่งเซกเมนต์ใดๆ จะต้องรอการตอบจาก cc2 ก่อนที่จะส่งเซกเมนต์ถัดไปได้ ช่วงที่ cc1 รอการตอบรับก็จะเป็นการช่วงหยุดที่ไม่สามารถนำส่งข้อมูลได้ และเท่ากับเป็นการใช้สายสื่อสารแบบฮาล์ฟดูเพล็กซ์ ซึ่งสูญเสียแบนด์วิดธ์ไปครึ่งหนึ่ง เนื่องจากต้องรอให้อีกด้านจัดส่งเสร็จสิ้นก่อน การตอบรับโดยไม่มีเทคนิคการเลื่อนช่องหน้าต่างจึงทำให้เสียประสิทธิภาพการถ่ายโอน ข้อมูล

ทีซีพีแก่ปัญหานี้โดยทำงานแบบฟูลดูเพล็กซ์ ฝ่ายส่งสามารถส่งเซกเมนต์ไปได้หลายเซกเมนต์โดยไม่ต้องรอการตอบรับทุกๆเซกเมนต์ แต่ทีซีพีต้องดำเนินการควบคุมปริมาณการส่งนี้ เพราะบัฟเฟอร์ฝ่ายรับย่อมมีขนาดจำกัด ทีซีพีฝ่ายรับจะต้องแจ้งให้ทางฝ่ายส่งถึงขนาดข้อมูลที่รับได้ โดยใช้เลขตอบรับ (ฟิลด์ acknowledgment number) และขนาดหน้าต่าง (ฟิลด์ window) สองค่านี้จึงใช้เป็นตัวกำหนดขนาดการรับส่งข้อมูลแต่ละครั้ง

รูปที่ 2.15 เป็นตัวอย่างทีซีพีเฮดเดอร์ซึ่งประกาศหน้าต่างขนาด 1024 ไบต์ ค่าในฟิลด์บ่งบอกว่าสถานีอีกด้านได้รับข้อมูลตั้งแต่ต้นถึงลำดับ 999 ครบถ้วนแล้ว และพร้อมรับข้อมูลไม่เกินเลขลำดับ 2023 (คำนวณจาก 1000-1+1024)

0 15 16 31

Source port			destination port
sequence number			
Acknowledgment number :1000			
Offset	Reserved	Code	Window size : 1024
Checksum			Urgent pointer
Options			
data			

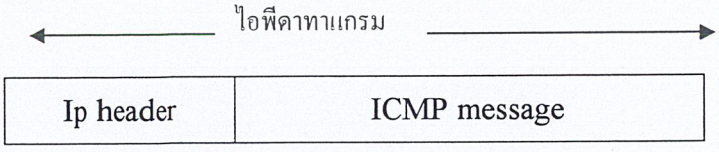
รูปที่ 2.15 ฟิลด์ในทีซีพีเฮดเดอร์ที่ใช้ควบคุมกระแสข้อมูล

2.6 ไอซีเอ็มพี และการตรวจสอบเส้นทาง (ICMP & Trace route)

เทร์ซเราท์ (Trace route) เป็นเครื่องมือที่ใช้ในการตรวจสอบเส้นทางการส่งข้อมูลของเครือข่ายทีซีพี/ไอพี โดยอาศัยไอซีเอ็มพี (ICMP : Internet Control Message Protocol) ทำให้ทราบถึงเส้นทางของข้อมูลว่ามีการผ่านเครือข่ายใดบ้าง ซึ่งข้อมูลเหล่านี้สามารถนำไปใช้ในการปรับปรุงประสิทธิภาพของระบบเครือข่ายได้

2.6.1 ไอซีเอ็มพี

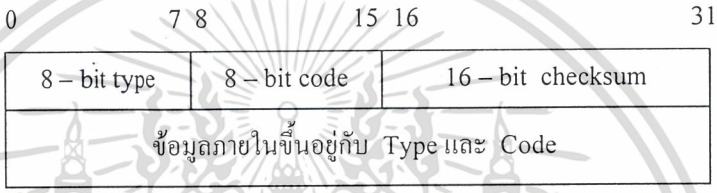
ไอซีเอ็มพีเป็นโปรโตคอลหนึ่งที่อยู่ใน TCP/IP Suite มีหน้าที่ส่งข่าวสารและคำสั่งควบคุมของ ไอพี โดยเฉพาะการรับ และ ส่ง Error Message ซึ่ง ไอซีเอ็มพีใช้ในการตรวจสอบสถานะและแจ้งของโฮสท์หรืออุปกรณ์ในเครือข่าย ทีซีพี/ไอพี โดยรูปแบบของไอพีเดทาแกรม (IP Datagram) เป็นดังนี้



รูปที่ 2.16 แสดงไอพีเดทาแกรม

ส่วนของ ICPM message จะประกอบด้วย

- type ขนาด 8 บิต : กำหนดทั้งค่าความผิดพลาดและการรายงานสถานะ การใช้งานในปัจจุบันมีทั้งหมด 15 ประเภท
- code ขนาด 8 บิต: รหัสความผิดพลาดย่อย
- checksum ขนาด 16 บิต : ค่าผลรวมตรวจสอบแบบ 1' complement สำหรับใช้ตรวจสอบความผิดพลาด โดยคำนวณผลรวมของ Type , code และ content
- Content ซึ่งมีขนาดไม่คงที่แตกต่างกันไปตาม type และ code ซึ่งฟิลด์นี้ใช้บรรจุข้อมูลข่าวสารเพิ่มเติมเพื่อแจ้งกลับ ดังรูป



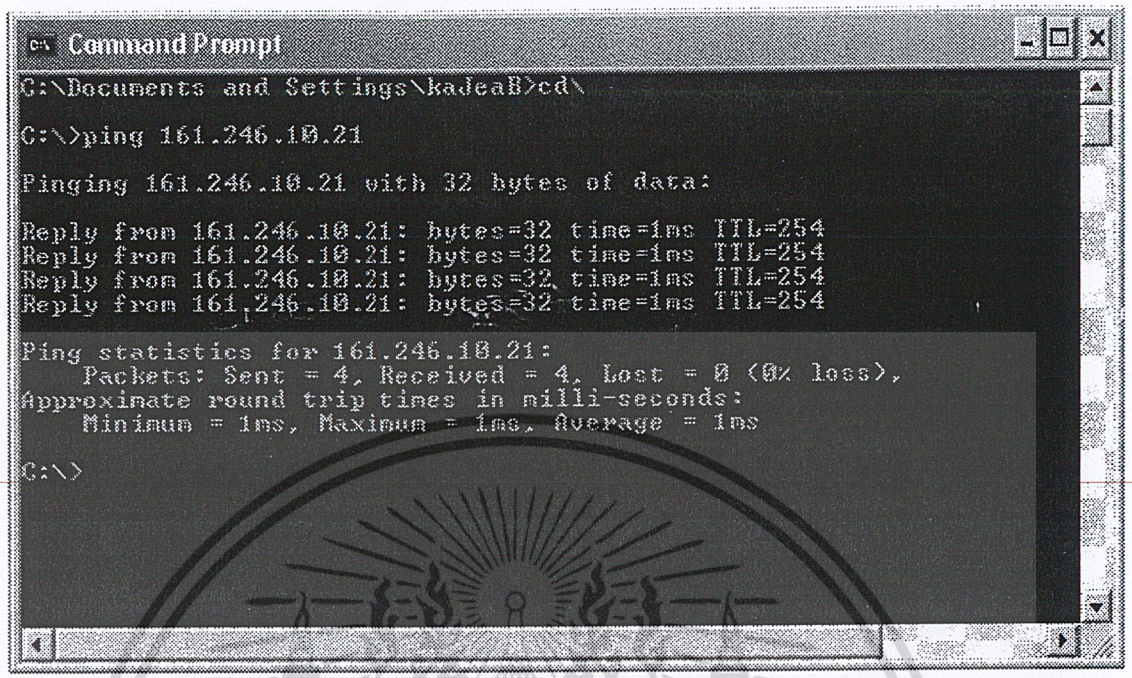
ความหมายของข้อมูลใน ICMP Message

บิตที่ 0-7 (ไบต์ที่ 0) ICMP Type เป็นฟิลด์ขนาด 8 บิตบอกถึงประเภทของ ICMP ที่กำลังสื่อสารอยู่

บิตที่ 8- 15 (ไบต์ที่ 1) ICMP Code เป็นฟิลด์ขนาด 8 บิตที่เก็บข้อมูลรหัสของ ICMP Message

บิตที่ 16-31 (ไบต์ที่ 2-3) checksum ใช้เป็นตัวตรวจสอบความถูกต้องของ ICMP Message ทั้งหมด

1 การใช้งาน ไอซีเอ็มพีสามารถนำไปใช้ได้หลายรูปแบบ เช่น การตรวจสอบสถานะของโฮสท์หรืออุปกรณ์โดยการส่ง ICPM type 8 เพื่อสอบถามไปยังปลายทาง ซึ่งเมื่อปลายทางได้รับก็จะตอบกลับมาเป็น ICMP type 0 หรือถ้าแพคเกจ ICMP นั้นไม่สามารถไปถึงผู้รับได้ เราเตอร์หรืออุปกรณ์ที่ได้รับข้อมูลจะทำการสร้าง ICMP type 3 กลับมา



รูปที่ 2.18 แสดงการใช้ ping ส่งไอซีเอ็มพีชนิด 8 ไปยัง 161.246.10.21

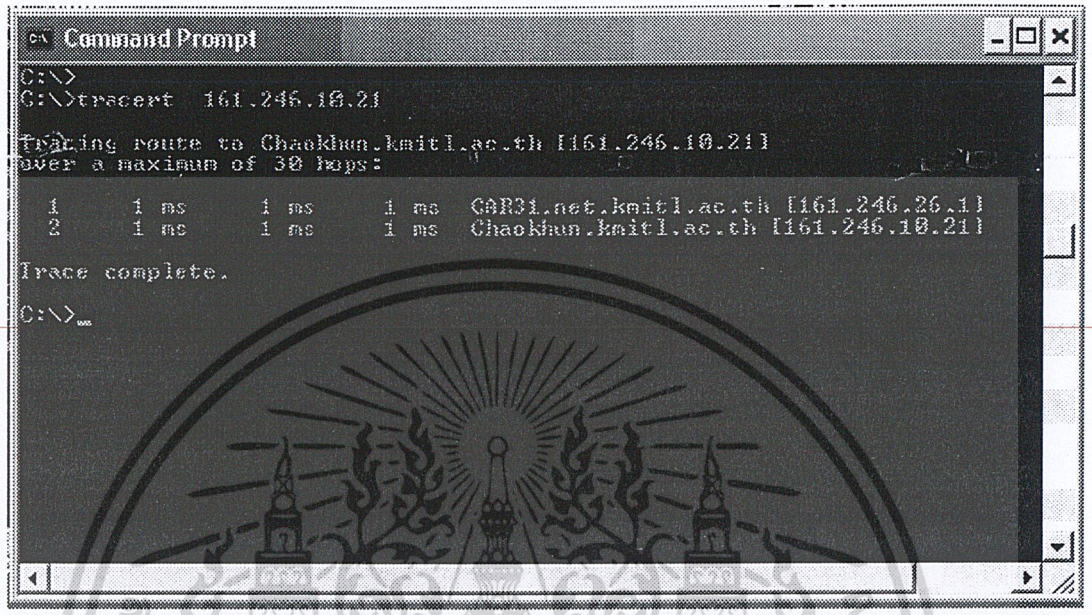
2 ในกรณีที่โฮสต์ส่งแพคเกจออกมาเร็วกว่าที่บัพเฟอร์ของเราเตอร์จะรับได้ เราเตอร์จะทำการส่ง ICMP type 4 ไปยังโฮสท์นั้น เมื่อโฮสท์ได้รับข้อมูลก็หน่วงเวลาการทำงานในการส่งข้อมูลให้ช้าลง

3 ใช้ในการอัพเดท routing table เมื่อเน็ทเวิร์กมีการเปลี่ยนแปลง เช่น เส้นทางในการเชื่อมต่อขาด หรือพบเส้นทางใหม่ที่ดีกว่า เราเตอร์จะทำการส่ง ICMP type 5 ไปยังโฮสท์หรือเราเตอร์ข้างเคียง เพื่อบอกการเปลี่ยนเส้นทาง

2.6.2 การตรวจสอบเส้นทาง (Trace route)

ในการทำงานของเทร์ซเราท์ จะสร้างแพคเกจติดต่อไปยังพอร์ทของโฮสท์หรืออุปกรณ์และเลือกหมายเลขพอร์ทที่เป็นไปไม่ได้ เช่น พอร์ท 30000 โดยในครั้งแรกจะกำหนดให้ค่า TTL = 1 เมื่อแพคเกจนี้ส่งไปยังเราเตอร์ตัวแรกหรือเกตเวย์ ค่าของ TTL จะเป็น 0 ซึ่งมีผลให้อุปกรณ์นั้นสร้าง ICMP type 11 code 0 (time exceeded) กลับมายังผู้ส่ง ผู้ส่งก็จะส่งแพคเกจอีกครั้งโดยกำหนด TTL = 2 ซึ่งจะเกิดกรณีของ time exceeded ขึ้นที่อุปกรณ์ตัวที่สอง การทำงานจะวนเวียนในลักษณะเช่นนี้โดยเพิ่มค่าของ TTL ทีละ 1 และเมื่อแพคเกจถูกส่งไปถึงปลายทาง ก็จะพยายามติดต่อปลายทางตามที่พอร์ทกำหนด ซึ่งปลายทางจะสร้าง ICMP type 3 code 3 (port

unreachable) กลับมายังผู้ส่ง ด้วยลักษณะการทำงานดังกล่าวมาทำให้สามารถรู้เส้นทางในการส่งข้อมูลได้



รูปที่ 2.19 แสดงการตรวจสอบเส้นทางโดยการเทร์ซเราท์ ไปที่ 161.246.10.21

ข้อดีของ การเทร์ซเราท์แบบนี้คือ เราเตอร์ทุกตัว มีความสามารถในการส่ง TTL message อยู่แล้ว จึงไม่ต้องพัฒนาส่วนอื่นเพิ่ม ส่วนข้อเสียคือ จำนวนแพคเกจที่ผลิตขึ้นนั้น มีมากถึง 2n โดย n คือจำนวน hop และเวลาที่เสียไป จากการผ่านซ้ำของ hop เดิมที่เคยผ่านไปแล้ว อีกประการหนึ่งคือ เส้นทางอาจมีการเปลี่ยนแปลงระหว่างการทำงานได้ และประการสุดท้าย คือ แนวคิดนี้ไม่ได้ตรวจสอบถึงเส้นทางขากลับ ซึ่งอาจแตกต่างจากเส้นทางขาไป

ตารางที่ 2.5 ประเภทของความผิดพลาดของไอซีเอ็มพี

Type	ความหมาย	การใช้งาน
0	Echo reply	แจ้งตอบรับกลับ
3	Destination unreachable	ไม่สมารถติดต่อปลายทางได้
4	Source quench	ให้ต้นทางลดระดับภาระงาน
5	Redirect	แจ้งเส้นทางที่เหมาะสมกว่า
8	Echo request	ร้องขอการตอบรับ
9	Router advertisement	เราเตอร์แจ้งประกาศตัวเอง
10	Router solicitation	โฮสต์ตรวจหาเราเตอร์
11	Time exceeded for datagram	เดทาแกรมใช้เวลาเกินกำหนด
12	Parameter problem on datagram	มีปัญหาพารามิเตอร์ในเดทาแกรม
13	Time stamp request	ร้องขอเวลาระบบ
14	Time stamp reply	แจ้งเวลาระบบกลับ
15	Information request	ร้องขอข่าวสาร
16	Information reply	แจ้งข่าวสารกลับ
17	Address mask request	ร้องขอแอดเดรสมาสค์
18	Address mask reply	แจ้งแอดเดรสมาสค์กลับ

การเทร์ซเราท์ ที่พัฒนาขึ้นใหม่เพื่อแก้ไขปัญหาของ การเทร์ซเราท์ เดิม โดยมีการออกแบบ ICMP message ใหม่ที่ออกแบบมาเฉพาะ โดยจะเป็น ไอซีเอ็มพีแบบ echo packet ซึ่งจะส่งไปยังโฮสต์หรืออุปกรณ์ปลายทาง โดยอุปกรณ์ที่ได้รับแพคเกจนี้จะทำการส่งต่อ และตอบกลับไปยังผู้ส่ง วิธีนี้ข้อดีคือ จำนวนแพคเกจจะมีเพียง n + 1 และการตรวจสอบเส้นทางสามารถทำได้เร็วขึ้น

F	C	number	length	ID number
Outbound Hop Count				Return Hop Count
Original IP Address				

รูปที่ 2.20 รูปแบบของ IP Trace route Option

รูปแบบของ IP Trace route Option มีรายละเอียดดังนี้

F ระบุว่าจะคัดลอกไป fragment หรือไม่

0 : ไม่คัดลอกไป fragment

1 : คัดลอกไป fragment

C ระบุ class

2 : debugging & measurement

Number

18 : F + C + Number = 82

ID Number เป็นตัวเลขที่ตั้งขึ้น โดยตัวผู้ส่งแพคเกจ เพื่อใช้เป็นหมายเลขพอร์ทของ ICMP trace route

Original IP Address เป็นหมายเลข IP ของผู้ส่ง

Outboud Hop Count (OHC) จำนวน hop ของแพคเกจที่ส่งไปผ่าน

Return Hop Count (RHC) จำนวนของ hop ที่แพคเกจที่ส่งกลับผ่าน

2.7 การพัฒนาระบบเชิงวัตถุ

การพัฒนาระบบซอฟต์เวร์ในอดีตนั้นจะใช้วิธีการพัฒนาแบบโครงสร้าง (structural) ซึ่งการพัฒนาระบบเชิงวัตถุนั้นประกอบด้วยขั้นตอนสองขั้นตอนที่สำคัญคือ ขั้นตอนของการวิเคราะห์และการออกแบบเชิงวัตถุ (Object-Oriented Analysis and Design)

การพัฒนาระบบเชิงวัตถุเป็นวิธีการพัฒนาในรูปแบบใหม่ ซึ่งการพัฒนาระบบด้วยวิธีเชิงวัตถุนั้น ผู้พัฒนาระบบจะมองระบบและส่วนประกอบหรือระบบย่อย (Sub System) เป็นวัตถุ (Object) เนื่องจากสิ่งต่างๆ ที่มีอยู่ในโลกความจริงนั้นเชื่อถือได้ว่าเป็นวัตถุอยู่แล้ว และด้วยการมองระบบเป็นวัตถุทำให้ผู้พัฒนาระบบสามารถมองระบบได้อย่างชัดเจนมากขึ้น ซึ่งส่งผลให้การวิเคราะห์และออกแบบระบบทำได้ง่ายและมีความถูกต้องมากขึ้น ในการวิเคราะห์และออกแบบเชิงวัตถุนั้นผู้พัฒนาระบบจะมองภาพเป็น ยูสเคส (Use Case) ซึ่งก็คือการมองฟังก์ชันการทำงานหลักของระบบ และ รายละเอียดการทำงานตามยูสเคส จากนั้นผู้พัฒนาระบบก็จะออกแบบและทำการพัฒนาระบบนี้ โดยจะเขียนออกมาเป็นแผนภาพยูสเคส, แผนภาพแสดงลำดับขั้นตอนการทำงาน (Sequence Diagram), แผนภาพการทำงานร่วมกันของวัตถุ (Collaboration Diagram), แผนภาพวัตถุ, แผนภาพคลาส และ แผนภาพสถานะ (State Diagram) โดยหลังจากที่ผู้พัฒนาระบบได้ทำการวิเคราะห์และออกแบบเป็นที่เรียบร้อยแล้วก็จะเข้าสู่การพัฒนาระบบ ซึ่งในขั้นตอนของการพัฒนาระบบนี้ก็จะมีภาษาในการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming Language) ซึ่งเป็นภาษาในการเขียนโปรแกรมที่สนับสนุนการสร้างคลาสและวัตถุ ส่งผลให้สิ่งที่ได้จากการวิเคราะห์และออกแบบนั้นสามารถนำมาใช้ในการเขียนโปรแกรมได้

ข้อดีของการพัฒนาระบบเชิงวัตถุ

การพัฒนาระบบเชิงวัตถุนั้นมีข้อดีในการที่จะนำมาใช้ในการพัฒนาดังต่อไปนี้

1. นักพัฒนาสามารถโมเดลระบบได้อย่างครบถ้วนมากยิ่งขึ้น

ปัญหาหนึ่งในการโมเดลระบบแบบโครงสร้าง (Structural Model) คือ การเปลี่ยนแปลงจากมุมมองของการวิเคราะห์เป็นการออกแบบนั้นทำได้ยาก ถึงแม้ว่ามุมมองทั้งสองจะมาจากระบบเดียวกัน แต่ว่าการโมเดลระบบแบบโครงสร้างทั้งสองไม่ได้แสดงความสัมพันธ์ระหว่างการวิเคราะห์ (แผนภาพ Data Flow และแผนภาพเอ็นติตี้รีเลชันชีพ (Entity Relationship)) ซึ่งนักพัฒนาระบบสามารถที่จะทำการออกแบบจากการวิเคราะห์ได้หลายรูปแบบ โดยแต่ละรูปแบบอาจมีลักษณะการทำงานที่ไม่เหมือนกันก็ได้ และในการที่จะทำการเปลี่ยนจากแผนภาพ Data Flow ที่เปลี่ยนเป็นโค้ดนั้นยังไม่มีหลักการที่แน่นอนในการเปลี่ยน ด้วยเหตุนี้โปรแกรมที่เขียนขึ้นมาอาจจะไม่ตรงตามที่ได้จากการวิเคราะห์ก็ได้

การพัฒนาระบบเชิงวัตถุนั้นเนื่องจากมีการใช้โมเดลโดยตลอดในการทำการวิเคราะห์และออกแบบ การกำหนดคลาสและวัตถุในโมเดลที่ได้จากวิเคราะห์นั้นสามารถที่จะเปลี่ยนเป็นโค้ดได้โดยตรง ซึ่งผู้พัฒนาระบบสามารถมองเห็นความสัมพันธ์ระหว่างการกำหนดปัญหาและวิธีการแก้ปัญหาแล้วลงมือพัฒนาได้อย่างง่ายขึ้น

2. เพิ่มความเข้าใจในโดเมนของปัญหา

เนื่องจากโมเดลนี้จำลองมาจากโลกจริงๆ การแยกแยะเอกลักษณ์ของการพัฒนาเชิงวัตถุนั้นให้ความเข้าใจและมีความสามารถที่ดีกว่า รวมทั้งมีการตั้งชื่อสำหรับวัตถุนั้นตั้งตามการใช้งานจริง ส่งผลให้มีความเข้าใจที่ตรงกันทั้งผู้ใช้งานและนักพัฒนาโปรแกรมเอง ซึ่งการโมเดลในการพัฒนาระบบเชิงวัตถุนั้นแทบไม่มีมุมมองของทางด้านคอมพิวเตอร์เข้าไปเกี่ยวข้องเลย ส่งผลให้ระบบที่ได้รับการพัฒนามีการแยกแยะระหว่างสิ่งที่เป็นอิสระต่อกัน

3. เพิ่มเสถียรภาพของการเปลี่ยนแปลง

นักพัฒนาระบบส่วนใหญ่นั้นเมื่อมีการเปลี่ยนแปลงความต้องการนั้นจำต้องมีการแก้ไขโครงสร้างของระบบ เนื่องจากว่าการออกแบบระบบแบบโครงสร้างนั้นมีพื้นฐานของระบบที่ไม่เสถียรและไม่คงทนต่อการเปลี่ยนแปลง

เนื่องจากการแยกแยะเอกลักษณ์ของการพัฒนาระบบด้วยวิธีเชิงวัตถุนั้นมีพื้นฐานมาจากโลกความจริง ดังนั้นระบบจึงมีความเสถียรมากกว่า การเปลี่ยนแปลงความต้องการในระบบก็เพียงแต่เพิ่มวัตถุหรือเปลี่ยนแปลงภายในวัตถุ

4. โมเดลที่สนับสนุนการนำกลับมาใช้ใหม่

5. สนับสนุนการปรับเปลี่ยนขนาดของระบบ (Scalability)

6. สนับสนุนการออกแบบระบบที่เชื่อถือได้และมีความปลอดภัย (Reliability and Safety)

7. สนับสนุนการทำงานแบบพร้อมกัน (Concurrency)

นักพัฒนาที่ใช้วิธีในการพัฒนาระบบเชิงวัตถุอาจจะไม่ได้รับประโยชน์จากวิธีนี้โดยทันทีทันใดแต่ว่านักพัฒนาจะค่อยๆ เห็นข้อดีของการพัฒนาระบบด้วยวิธีเชิงวัตถุ โดยเฉพาะอย่างยิ่งกับระบบที่มีความซับซ้อนและมีความต้องการไม่ชัดเจน

2.7.1 ความหมายของคำว่าออปเจคท์

ออปเจคท์ หรือ วัตถุ คือ สิ่งต่างๆ ที่อยู่รอบๆ ตัวเรา ซึ่งออปเจคท์หนึ่งอาจจะประกอบด้วยออปเจคท์อื่นๆหลายๆ ออปเจคท์ โดยจะมีขนาดที่เล็กที่สุดหรือใหญ่ที่สุดของออปเจคท์จะขึ้นอยู่กับขอบเขตของระบบที่เรากำลังสนใจอยู่ว่าต้องการรายละเอียดในระดับใด

ในการกำหนดออปเจคท์จะต้องกำหนดคลาสขึ้นมาก่อน คลาสคือรูปแบบทั่วไปของออปเจคท์ หรือเป็นพิมพ์เขียวของออปเจคท์และออปเจคท์คือ อินสแตนซ์ (instance) ของคลาส

ออปเจคท์มีโครงสร้างที่ประกอบด้วยแอทริบิวท์ (ข้อมูล) และพฤติกรรม (behavior) หรือโอเปอร์เรชัน หรือเมธอด ซึ่งแอททริบิวท์และพฤติกรรมดังกล่าวจะมีอะไรได้บ้างนั้น จะขึ้นอยู่กับระบบที่ต้องการออกแบบ โดยในการออกแบบระบบจำนวนของแอทริบิวท์และโอเปอร์เรชันของ
ออบเจคท์นั้นจะขึ้นอยู่กับขอบเขตของระบบที่เราต้องการออกแบบ

2.7.2 หลักการของการพัฒนาเชิงวัตถุ

1. การแยกแยะเอกลักษณ์ (Abstraction)

การแยกแยะเอกลักษณ์ หมายถึง การกำหนดออบเจคท์ โดยไม่ได้ระบุรายละเอียดของแอททริบิวท์ (Attribute) และ พฤติกรรม (Behavior) หรือระบุเป็นเท่าที่สามารถทำได้ เนื่องจากความแตกต่างของมุมมองในการพิจารณาออบเจคท์ทำให้มีรายละเอียดที่แตกต่างกันเช่น

ในการกำหนดออบเจคท์ของเครื่องซักผ้า ในมุมมองของวิศวกรออกแบบ และในมุมมองของเจ้าของร้านรับซักรีดย่อมมีความแตกต่างกันแต่ก็ยังคงเป็นออบเจคท์เดียวกัน

การแยกแยะเอกลักษณ์ ทำให้ในการออกแบบมีความยืดหยุ่นมากขึ้น และไม่เกิดการกำหนดออบเจคท์เดียวกันมากกว่า 1 ออบเจคท์ ซึ่งผู้ใช้ในแต่ละฟังก์ชันสามารถกำหนดแอททริบิวท์และพฤติกรรมเพิ่มเติมให้กับออบเจคท์ตามความจำเป็นต้องมีในฟังก์ชันงานนั้นๆ

2. การสืบทอดคุณสมบัติ (Inheritance)

การสืบทอดคุณสมบัติ กล่าวคือ เมื่อออบเจคท์เป็นอินสแตนซ์ของคลาสแล้วออบเจคท์จะมีคุณสมบัติทั้งหมดของคลาสนั้นด้วย ซึ่งคลาสกลุ่มของออบเจคท์ชนิดเดียวกันออบเจคท์เป็นอินสแตนซ์ของคลาส และในการเขียนโปรแกรม คลาสคือต้นแบบที่ใช้ในการสร้างออบเจคท์ใหม่ ซึ่งไม่เพียงแต่ออบเจคท์เท่านั้นที่สืบทอด (inherit) จากคลาส แต่ว่าคลาสยังสามารถสืบทอดจากคลาสอื่นได้ด้วย และสามารถสืบทอดจากคลาสได้มากกว่า 1 คลาส (multiple inheritance)

คลาสที่สืบทอดจะเรียกว่า ซับคลาส (sub class) ส่วนคลาสที่ถูกสืบทอดนั้นจะเรียกว่า ซูปเปอร์คลาส (super class) ซึ่งซับคลาส จะมีคุณสมบัติเหมือนกับซูปเปอร์คลาสทุกประการโดยไม่ต้องกำหนดซ้ำ

แนวคิดในเรื่องการสืบทอดคุณสมบัติ นี้ มีประโยชน์ในการออกแบบระบบเพราะถ้ามีการออกแบบที่ดีแล้วจะสามารถนำคลาสที่ออกแบบไว้แล้วกลับมาใช้ใหม่ได้ (reusable) โดยการเพิ่มแอททริบิวท์หรือพฤติกรรมเพื่อให้เกิดเป็นออปเจคท์ใหม่โดยไม่ต้องสร้างตั้งแต่ต้น โดยเฉพาะด้านการเขียนโปรแกรมจะมองเห็นประโยชน์ของคุณสมบัตินี้ได้อย่างชัดเจน

3. โพลีมอร์พิซึม (Polymorphism)

คุณสมบัติโพลีมอร์ฟิซึมจะหมายถึงคลาสที่มีการสืบทอด แต่โอเปอร์เรชั่นในซับคลาส (subclass) มีการทำงานที่แตกต่างกันกับโอเปอร์เรชั่น (operation) เดียวกันในซับคลาส เช่น ถ้าคลาส Polygon มีโอเปอร์เรชัน draw และคลาส Rectangle สืบทอดจากคลาส polygon ซึ่งจะทำให้มีโอเปอร์เรชัน draw เช่นเดียวกัน แต่การ draw ของ Polygon กับ Rectangle มีการทำงานคนละอย่างกัน ดังนั้นจึงต้องมีการกำหนดโอเปอร์เรชัน draw ของคลาส Rectangle ใหม่

ลักษณะของโพลีมอร์ฟิซึม อีกรูปแบบหนึ่งคือ ภายในคลาสเดียวกันมีโอเปอร์เรชั่นชื่อเดียวกันมากกว่า 1 โอเปอร์เรชั่น โดยจะมีพารามิเตอร์ของโอเปอร์เรชันที่แตกต่างกัน โพลีมอร์ฟิซึมมีประโยชน์มากในการพัฒนาซอฟต์แวร์ โดยนักพัฒนาสามารถพัฒนาคลาสต่างๆ โดยมีโอเปอร์เรชันเดียวกัน แต่มีวิธีการทำงานที่แตกต่างกัน และสามารถทำการสืบทอด คลาสเดิมที่มีอยู่โดยแก้ไขเฉพาะโอเปอร์เรชันที่จำเป็นได้ทำให้ไม่ต้องมีการเปลี่ยนแปลงในคลาสที่มีอยู่แล้ว ซึ่งทำให้โปรแกรมอื่นๆ ที่ใช้งานคลาสนั้นอยู่ไม่มีผลกระทบใดๆ

4. การซ่อนรายละเอียด (Encapsulation)

คุณสมบัติการซ่อนรายละเอียด (Encapsulation) เป็นการซ่อนข้อมูลหรือกระบวนการทำโอเปอร์เรชันจากผู้ใช้คลาส เช่น ในคลาสโทรทัศน์นั้น ผู้ใช้ที่ดูการแสดงผลทางหน้าจอโทรทัศน์แต่ว่าจะไม่ทราบถึงวงจรอิเล็กทรอนิกส์ภายใน หรือกระบวนการทำงานต่างๆ ที่ทำให้เกิดภาพทางหน้าจอ ผู้ผลิตโทรทัศน์ (ผู้สร้างคลาส) จะกำหนดให้ผู้ใช้สามารถเปลี่ยนสถานี, ปรับเสียง, ปรับความคมชัดของภาพ และอื่นๆ เท่าที่จำเป็นเท่านั้น ซึ่งจะเรียกว่าส่วนอินเทอร์เฟส (interface) ของคลาส

ข้อดีของการซ่อนรายละเอียด คือทำให้สามารถป้องกันการใช้งานที่ผิดพลาด โดยการกำหนดให้ผู้ใช้คลาสสามารถใช้คลาสตามวิธีที่กำหนดไว้เท่านั้น ซึ่งจะจำกัดขอบเขตของปัญหาที่อาจเกิดขึ้นให้มีขนาดเล็กลง ข้อดีอีกประการหนึ่งคือเป็นการซ่อนการเปลี่ยนแปลงที่เกิดขึ้น ใน

กรณีที่มีการแก้ไขคลาส โดยผู้ใช้ไม่จำเป็นต้องทราบถึงการเปลี่ยนแปลงนั้น และยังได้ผลการทำงานในรูปแบบเดิม แต่ผลที่ได้อาจจะเปลี่ยนแปลงไปตามการเปลี่ยนแปลงของวิธีการ

5. การส่งแมสเสจ (Message Sending)

ระบบหนึ่งๆ ออบเจคท์สามารถทำงานร่วมกันได้ โดยการส่งแมสเสจออบเจคท์หนึ่งจะส่งแมสเสจให้กับอีกออบเจคท์หนึ่งเพื่อให้ออบเจคท์นั้นทำโอเปอร์เรชันอย่างใดอย่างหนึ่ง ออบเจคท์ที่ได้รับแมสเสจก็จะทำโอเปอร์เรชันตามแมสเสจที่ได้รับ

ความหมายของแมสเสจในที่นี้เป็นความหมายกว้างๆ ในการกำหนดให้ออบเจคท์ที่โอเปอร์เรชันที่ต้องการ เช่น การใช้ออบเจคท์ รีโมท (remote)ในการเปิดออบเจคท์โทรทัศน์ ออบเจคท์ รีโมทจะส่งแมสเสจในรูปแบบของคลื่นอินฟราเรดไปยังออบเจคท์ โทรทัศน์ซึ่งเมื่อออบเจคท์โทรทัศน์ได้รับแมสเสจนั้นก็จะทำโอเปอร์เรชันเปิดเครื่อง (turn-on) เพื่อเปิดเครื่อง

6. การเป็นส่วนหนึ่งของ (Aggregation)

เป็นรูปแบบความสัมพันธ์แบบหนึ่ง โดยออบเจคท์หนึ่งประกอบขึ้นจากออบเจคท์อื่นๆ ตั้งแต่ 1 ออปเจคท์ขึ้น โดยอาจจะเป็นความสัมพันธ์แบบเป็นส่วนหนึ่งของธรรมดา หรือ คอมโพซิชัน (Composition) ความสัมพันธ์แบบคอมโพซิชันเป็นรูปแบบหนึ่งของความสัมพันธ์แบบเป็นส่วนหนึ่งของ (Aggregation)

7. ความสัมพันธ์แบบแอซโซซิเอชัน (Associations)

เมื่อออปเจคท์หนึ่งใช้บริการของอีกออปเจคท์หนึ่งแต่ว่าไม่ได้เป็นเจ้าของวัตถุนั้น จะเรียกความสัมพันธ์นี้ว่า แอซโซซิเอชัน (Associations) ซึ่งความสัมพันธ์แบบนี้จะใช้ได้อย่างเหมาะสม ตัวอย่างเช่น การเรียกใช้บริการนั้นไม่ได้มีความสัมพันธ์แบบเป็นส่วนหนึ่งของ, ความสัมพันธ์นั้นเป็นแบบไคล์เอ็นท์ – เซิร์พเวอร์เป็นต้น ซึ่งความสัมพันธ์กับออบเจคท์หนึ่งในอีกคลาสหนึ่ง โดยจะมีได้ 2 แบบ คือ “หนึ่งต่อหนึ่ง” (one –to- one) และ “หนึ่งต่อหลายสิ่ง” (one -to-many)

2.7.3 ภาษาในการสร้างโมเดลในรูปแบบมาตรฐาน UML

ภาษาในการสร้างโมเดลในรูปแบบมาตรฐาน (Unified Modeling language : UML) เป็นผลงานที่เกิดจากความร่วมมือของ Grady Booch , Jame Rumbaugh และ Ivar Jacobson ซึ่งเป็นท่านผู้เชี่ยวชาญในด้านการพัฒนาระบบเชิงวัตถุ โดยการเข้าร่วมงานใน Rational Software Corporation ในปี 1994 ซึ่งในปี 1997 ซึ่งทาง OMG ได้มีการปรับปรุงในเวอร์ชันนี้ 2 ครั้ง (UML 1.2) และได้เริ่มมีการนำไปใช้ในอุตสาหกรรมต่างๆ อย่างกว้างขวาง

องค์ประกอบของ UML

UML เป็นภาษาโมเดลในรูปแบบกราฟฟิก ซึ่งใช้ประกอบกันเป็นไดอะแกรม โดยมีกฎในการประกอบกันของอิลิเมนต์ต่างๆ แผนภาพ (diagram) จะแสดงถึงมุมมองต่างๆ (multiple view) ของระบบ ซึ่งรวมเรียกว่า โมเดล UML จะบ่งบอกถึงรายละเอียดของระบบ แต่จะไม่ระบุถึงวิธีการในการพัฒนาระบบ (implementation) ใน UML นั้น มีไดอะแกรมอยู่ 9 ไดอะแกรม ดังต่อไปนี้

1. แผนภาพคลาส (Class Diagram)

ออบเจคท์ต่างๆ ในระบบจะประกอบด้วยแอททริบิวท์ (attribute) หรือ ป็อปเพอร์ตี (property) และ โอเปอร์เรชัน (หรือ พฤติกรรม) ออบเจคท์ที่มีแอททริบิวท์ และโอเปอร์เรชันเดียวกัน จะรวมเรียกว่าคลาส หรือกล่าวอีกนัยหนึ่งคือ คลาสจะเป็นต้นแบบหรือพิมพ์เขียวของออบเจคท์ และในทางกลับกันออบเจคท์คือ อินสเตนซ์ (instance) ของคลาส

2. แผนภาพวัตถุ (Object Diagram)

ออบเจคท์ คืออินสแตนท์ (instance) ของคลาส การกำหนดรายละเอียดของออบเจคท์ก็คือการกำหนดค่าให้กับแอททริบิวส์ เช่น ออบเจคท์ washing machine ชื่อ My Washer มี brand name คือ Laundatorium , model name คือ Washermeister, serial number คือ AD1145 และ capacity คือ 16 ปอนด์

3. แผนภาพยูสเคส (Use Case Diagram)

เป็นแผนภาพกิจกรรมที่ใช้ติดต่อระหว่างระบบกับผู้ใช้ซึ่งจะได้มาจากมุมมองของผู้ใช้ ซึ่งในแง่ของการพัฒนาระบบ ไดอะแกรมนี้มีความสำคัญมากในจุดเริ่มต้นในการกำหนดความต้องการของระบบ โดยพิจารณาจากมุมมองของผู้ใช้ ซึ่งจะทำให้ได้ระบบที่ตรงกับความต้องการและสามารถในไปใช้งานได้จริง

4. แผนภาพสถานะ (State Diagram)

เป็นแผนภาพที่ใช้แสดงพฤติการณ์ของระบบ กล่าวคือ การทำงาน ณ เวลาใดๆ ซึ่งออบเจคท์จะอยู่ในสถานะ (state) ใดสถานะหนึ่งเช่น elevator อยู่ในสถานะ moving upward, stopped หรือ moving downward หรือ washing machine มีสถานะคือ soak, wash, rinse, spin หรือ off เป็นต้น

5. แผนภาพกิจกรรม (Activity Diagram)

ใช้แสดงการไหลเวียนของอิเวนต์ (Event) ในยูสเคส กล่าวคือ เป็นแผนภาพกิจกรรมที่เกิดตามยูสเคสหรือเกิดจากพฤติกรรมของออบเจคท์เองตามปกติเป็นลำดับของกิจกรรม

6. แผนภาพการทำงานร่วมกันของวัตถุ (Collaboration Diagram)

สิ่งต่างๆ ในระบบ จะทำงานร่วมกันเพื่อทำให้เกิดผลตามจุดประสงค์ของระบบ รูปแบบการทำงานร่วมกันนี้จะแสดงโดยใช้แผนภาพ Collaboration

7. แผนภาพคอมโพแนนท์ (Component Diagram)

แผนภาพคอมโพแนนท์ ใช้แสดงโครงสร้างทางกายภาพของซอฟแวร์ ซึ่งเป็นแผนภาพสำหรับการพัฒนาซอฟต์แวร์ยุคใหม่ที่ใช้หลักการของคอมโพแนนท์ เช่น control ต่างๆ ใน Visual basic เป็นต้น

8. แผนภาพดีพอยเมนต์ (Deployment Diagram)

แผนภาพดีพอยเมนต์นั้นจะแสดงสถาปัตยกรรมของระบบคอมพิวเตอร์ และอุปกรณ์ในการเชื่อมต่อ ต่างๆ ไม่ว่าจะเป็นฮาร์ดแวร์และแสดงซอฟต์แวร์ที่ติดตั้งในแต่ละระบบ ซึ่งมีประโยชน์ในแง่ของการติดตั้ง

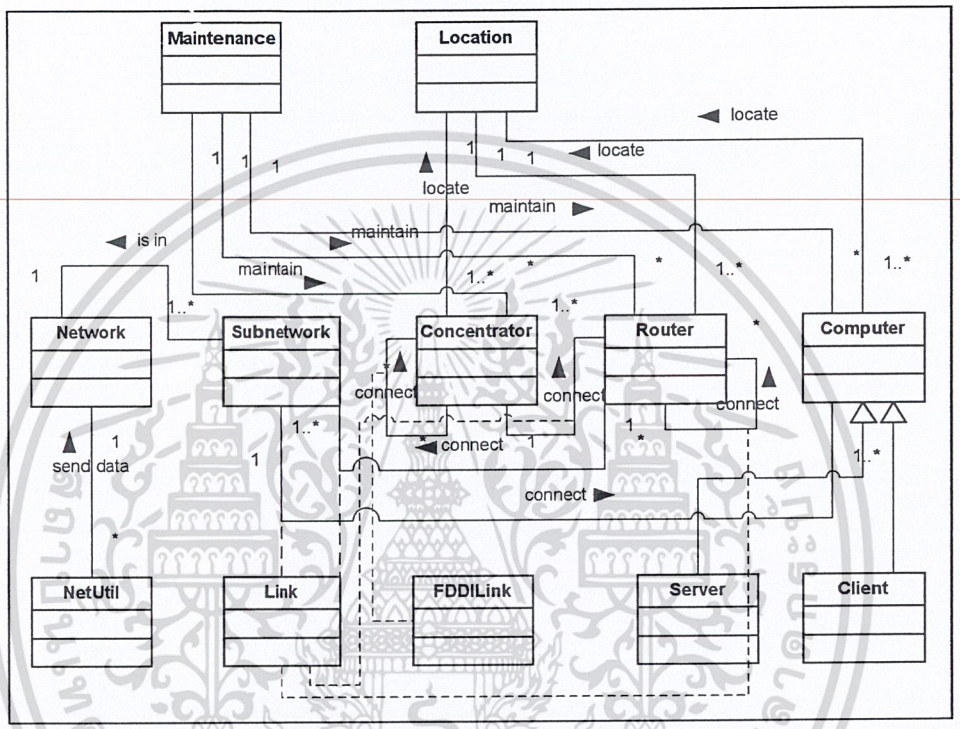
9. แผนภาพอื่นๆ

นอกจากไดอะแกรมทั้ง 9 ไดอะแกรมแล้ว UML ยังได้กำหนดคุณสมบัติอื่นๆ เพื่อเพิ่มความสามารถของไดอะแกรม โดยคุณสมบัติเหล่านี้สามารถใช้ได้เกือบทุกไดอะแกรม ขึ้นอยู่กับลักษณะของไดอะแกรมนั้น ได้แก่ แพ็กเก็ต (Package) ซึ่งใช้ในการรวมอิลิเมนต์ของไดอะแกรมให้เป็นกลุ่ม (group) , โน๊ต (Note) ซึ่งใช้ในการเพิ่มรายละเอียดที่ต้องการซึ่งในแผนภาพ เป็นต้น

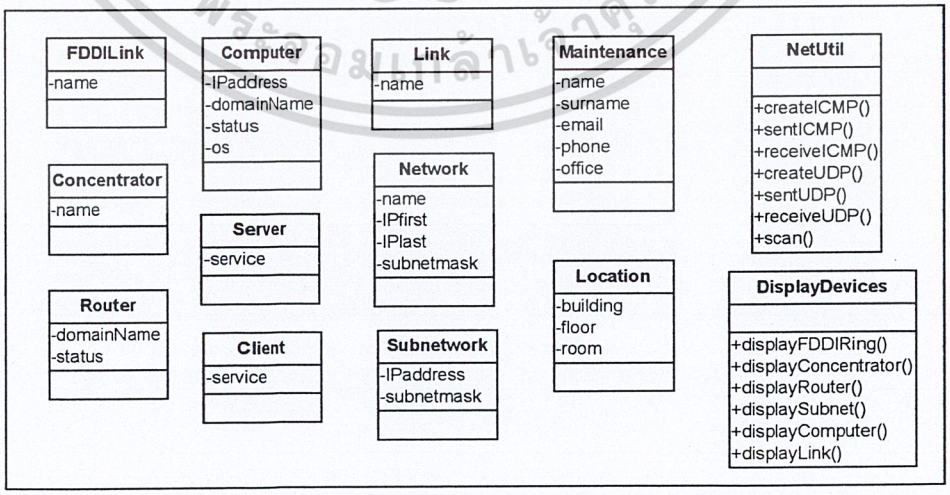
บทที่ 3

การออกแบบโปรแกรม

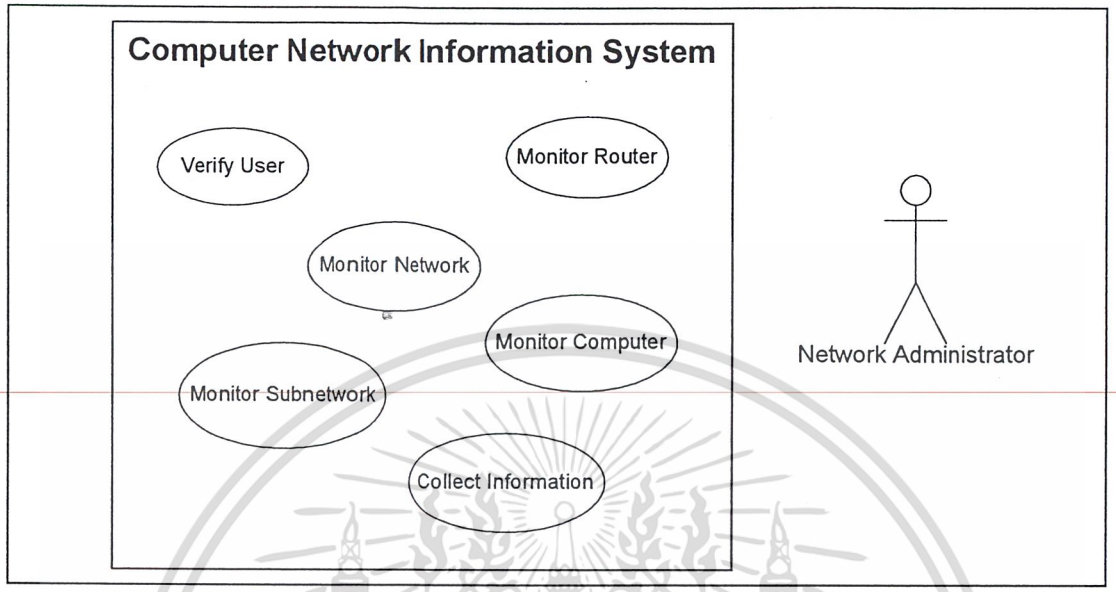
Client Interview



Class Diagram



High-Level Use Cases

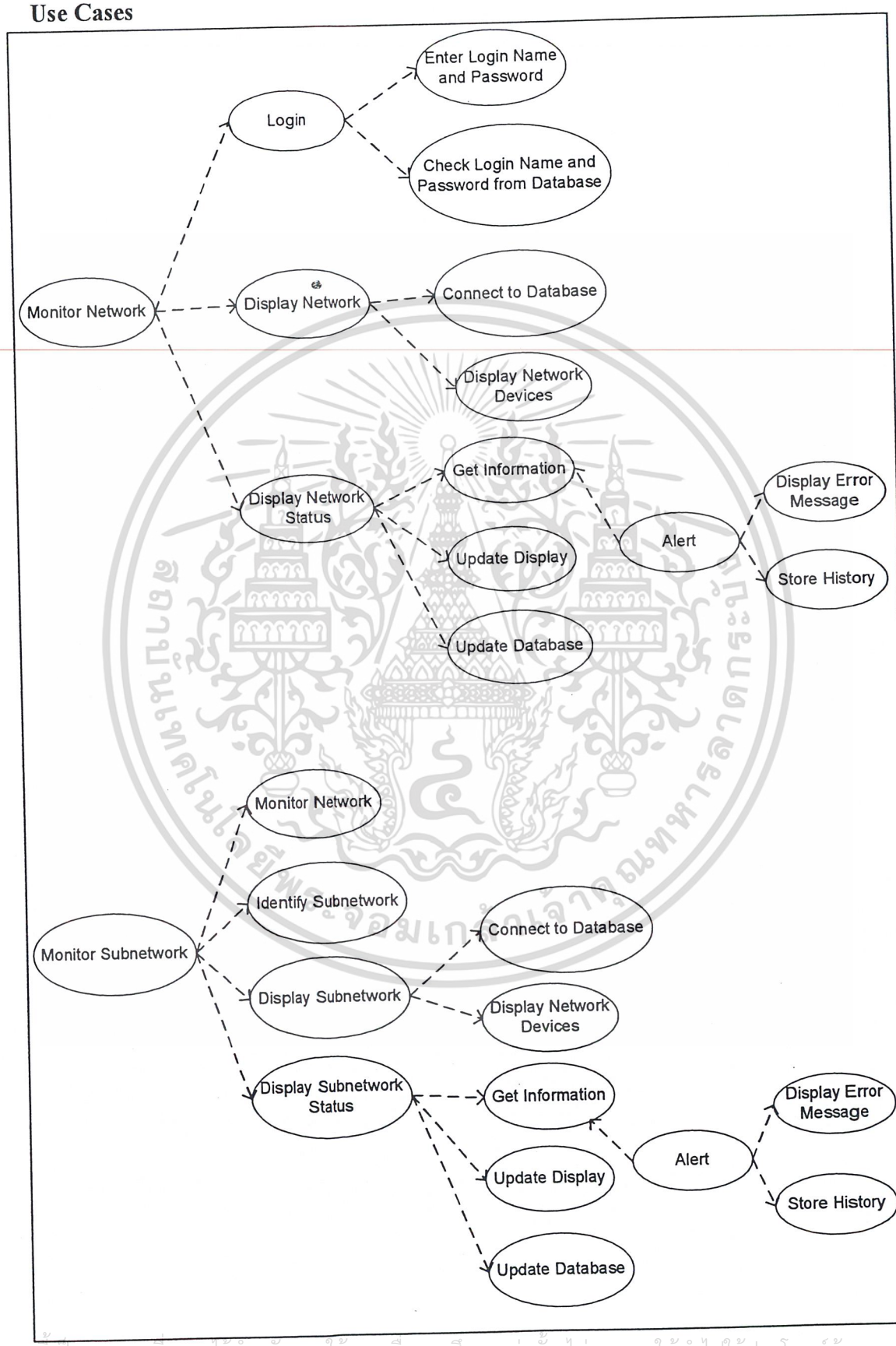


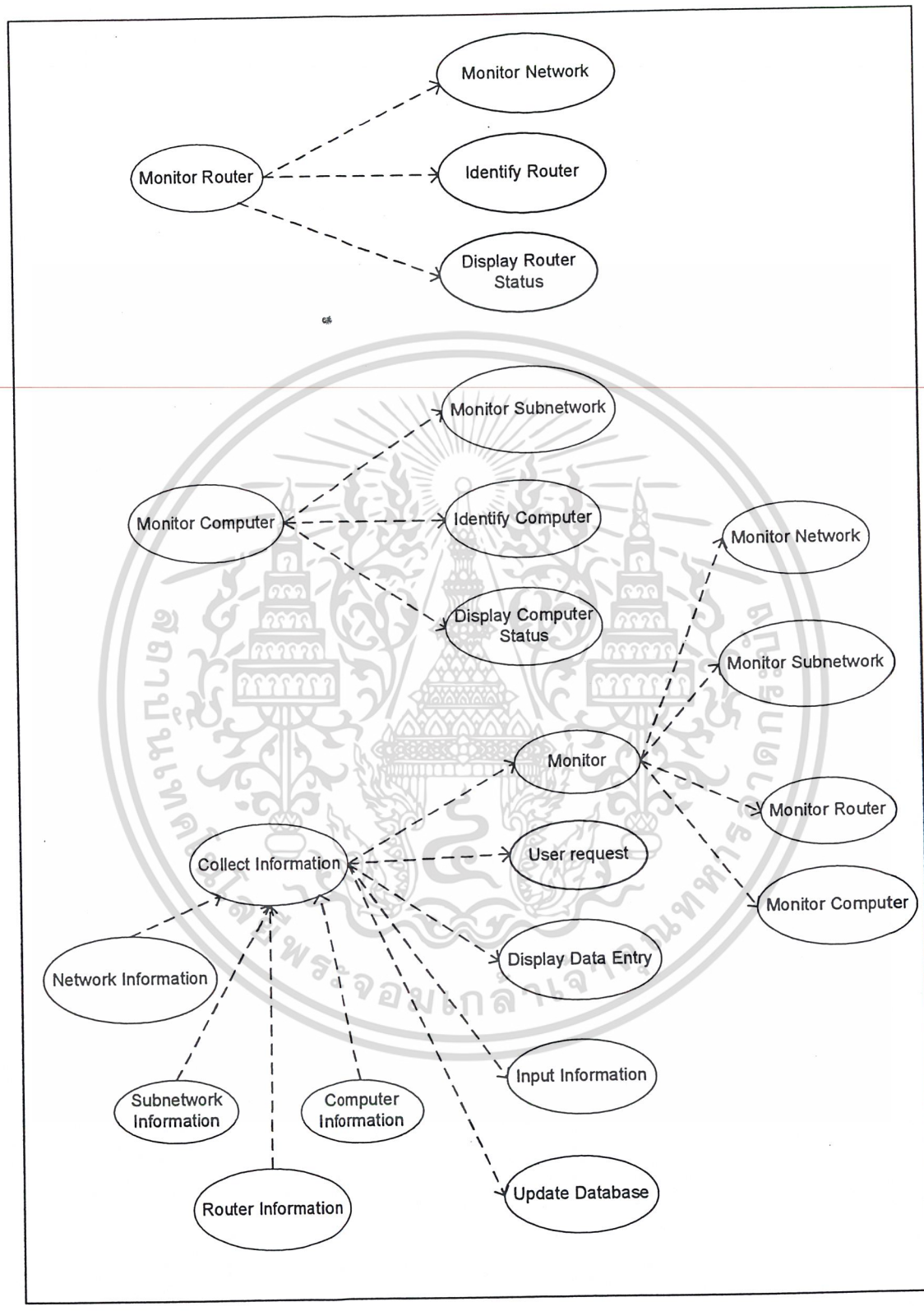
รูปที่ 3.3 แสดงแผนภาพ High-Level Use cases

หน้าที่การทำงานหลักของระบบมีดังต่อไปนี้

1. Verify User : ตรวจสอบผู้ใช้ระบบว่ามีความสามารถในการเข้าใช้ระบบหรือไม่
2. Monitor Network : แสดงแผนที่เครือข่ายหลักของสถาบันฯ ประกอบด้วย Backbone ที่เป็นแบบ FDDI (concentrator และ fiber optic link) ต่อเชื่อมกับ router และจาก router เชื่อมโยงไปยัง subnetwork ต่าง ๆ
3. Monitor Subnetwork : แสดงเครือข่ายย่อยของสถาบันฯ ประกอบด้วยคอมพิวเตอร์ที่อยู่ใน subnetwork ที่ระบุ
4. Monitor Router : ตรวจสอบสถานะการทำงานของ router ตัวที่ระบุว่าทำงานอยู่หรือไม่ , bandwidth utilization เป็นเท่าไหร่
5. Monitor Computer : ตรวจสอบสถานะการทำงานของคอมพิวเตอร์ตัวที่ระบุว่าเปิดหรือปิด , ให้บริการหรือใช้บริการอะไร , ใช้ระบบปฏิบัติการอะไร
6. Collect Information : เป็นการเก็บรวบรวมข้อมูลเพิ่มเติมที่ผู้ใช้สามารถใส่ลงไปในโปรแกรมได้ เช่น ผู้ดูแลอุปกรณ์ , สถานที่ตั้งของอุปกรณ์ , รายละเอียดอื่น ๆ

Use Cases

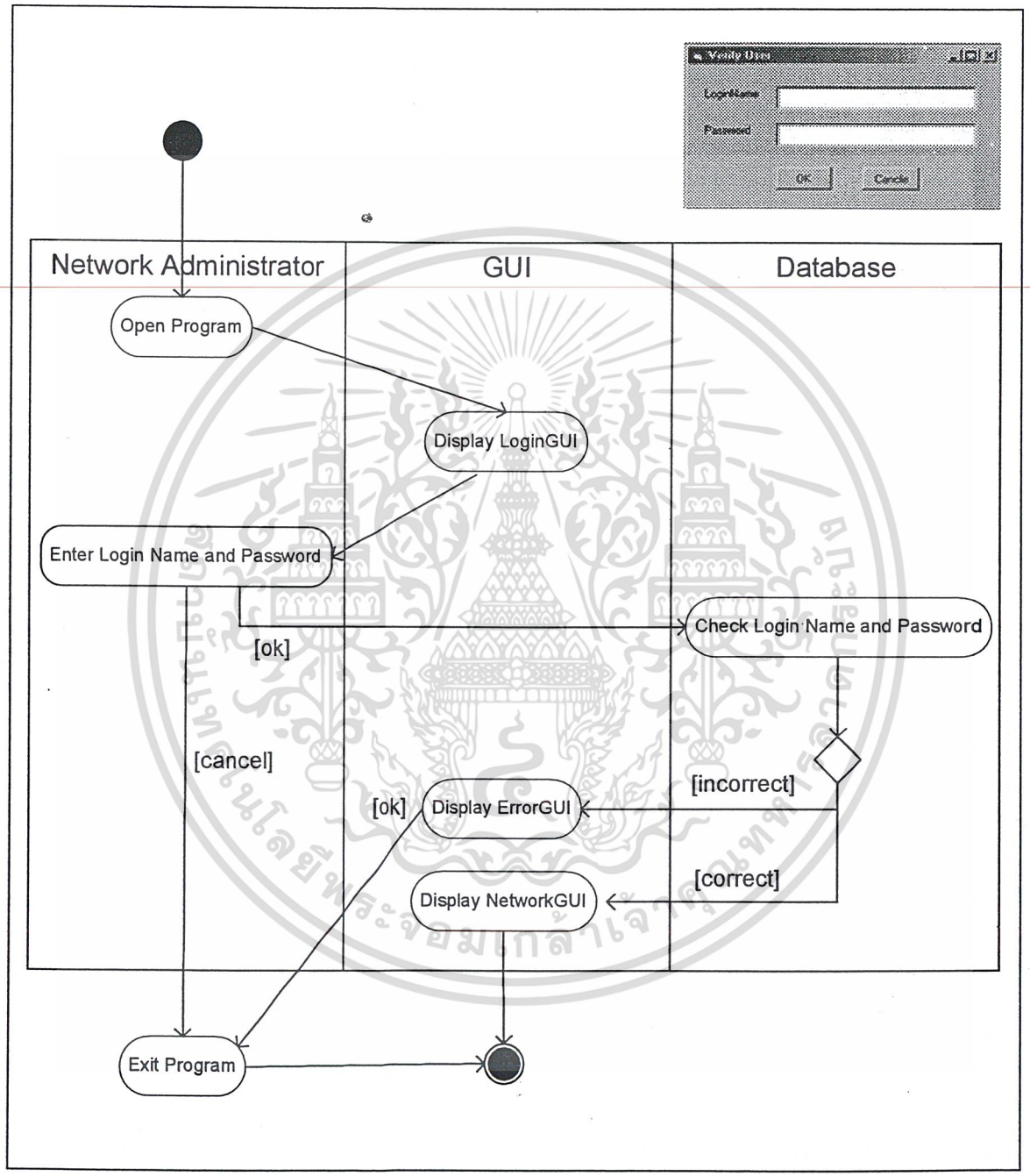




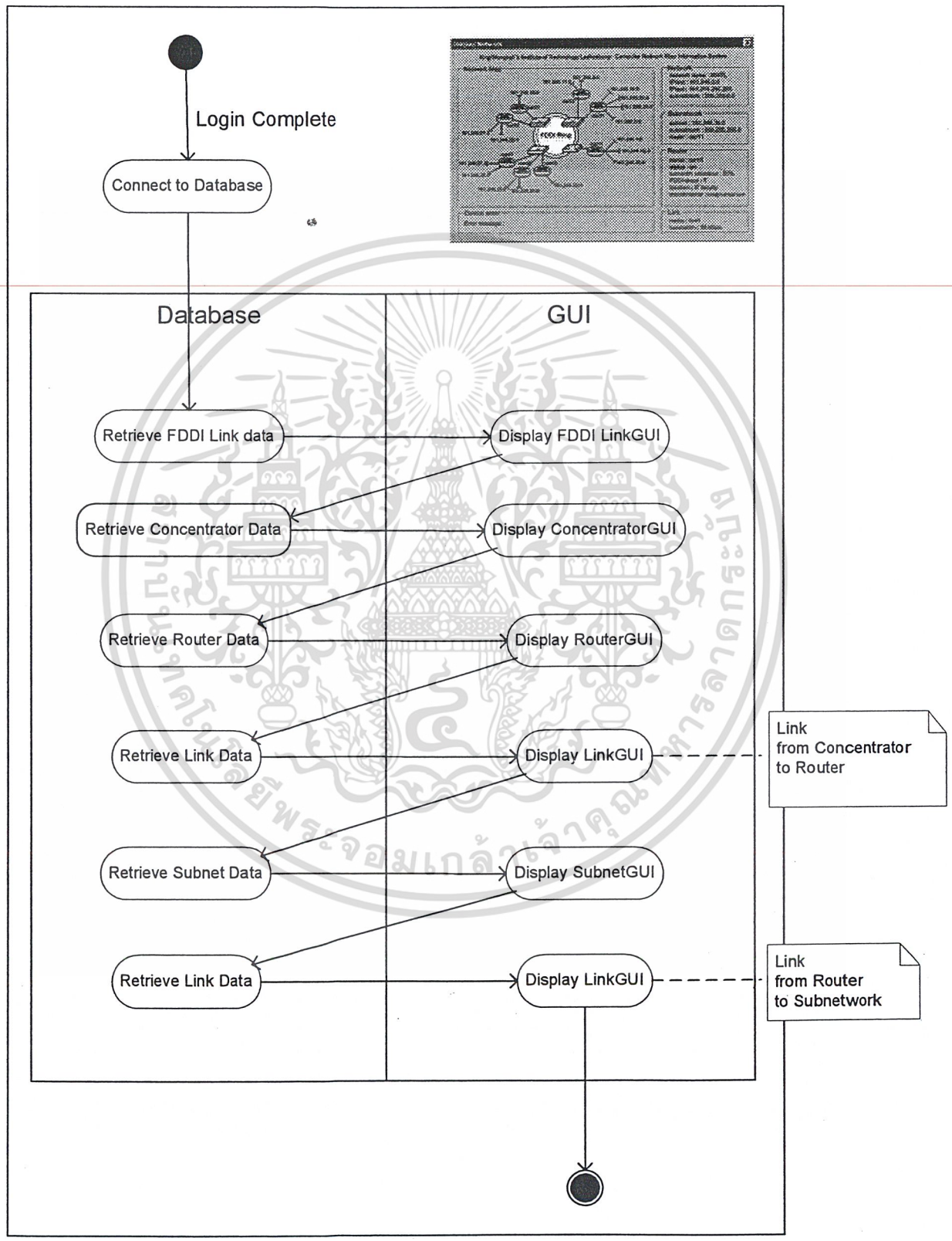
รูปที่ 3.4 แสดงแผนภาพ Use Cases

Activity Diagram

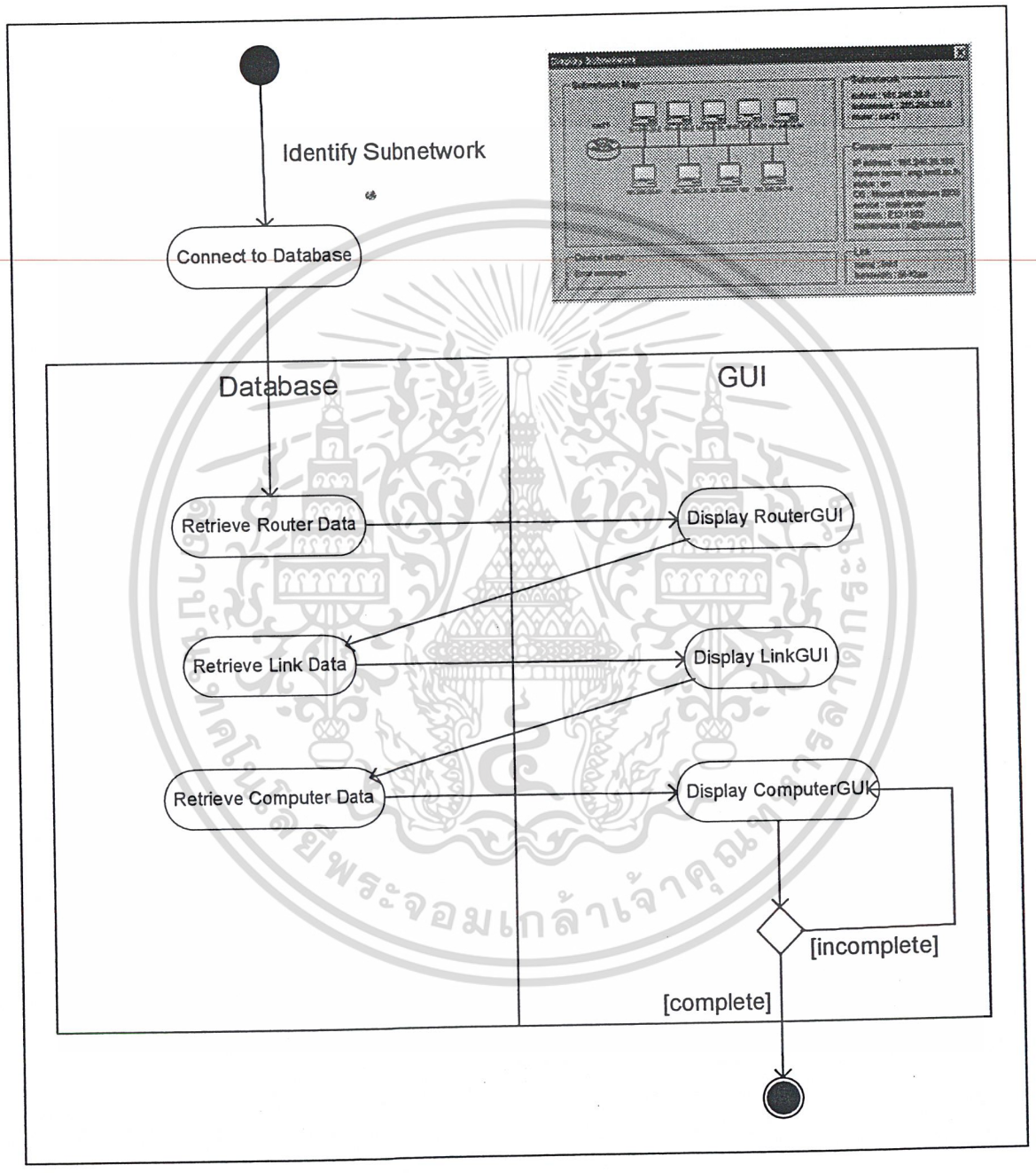
Verify User



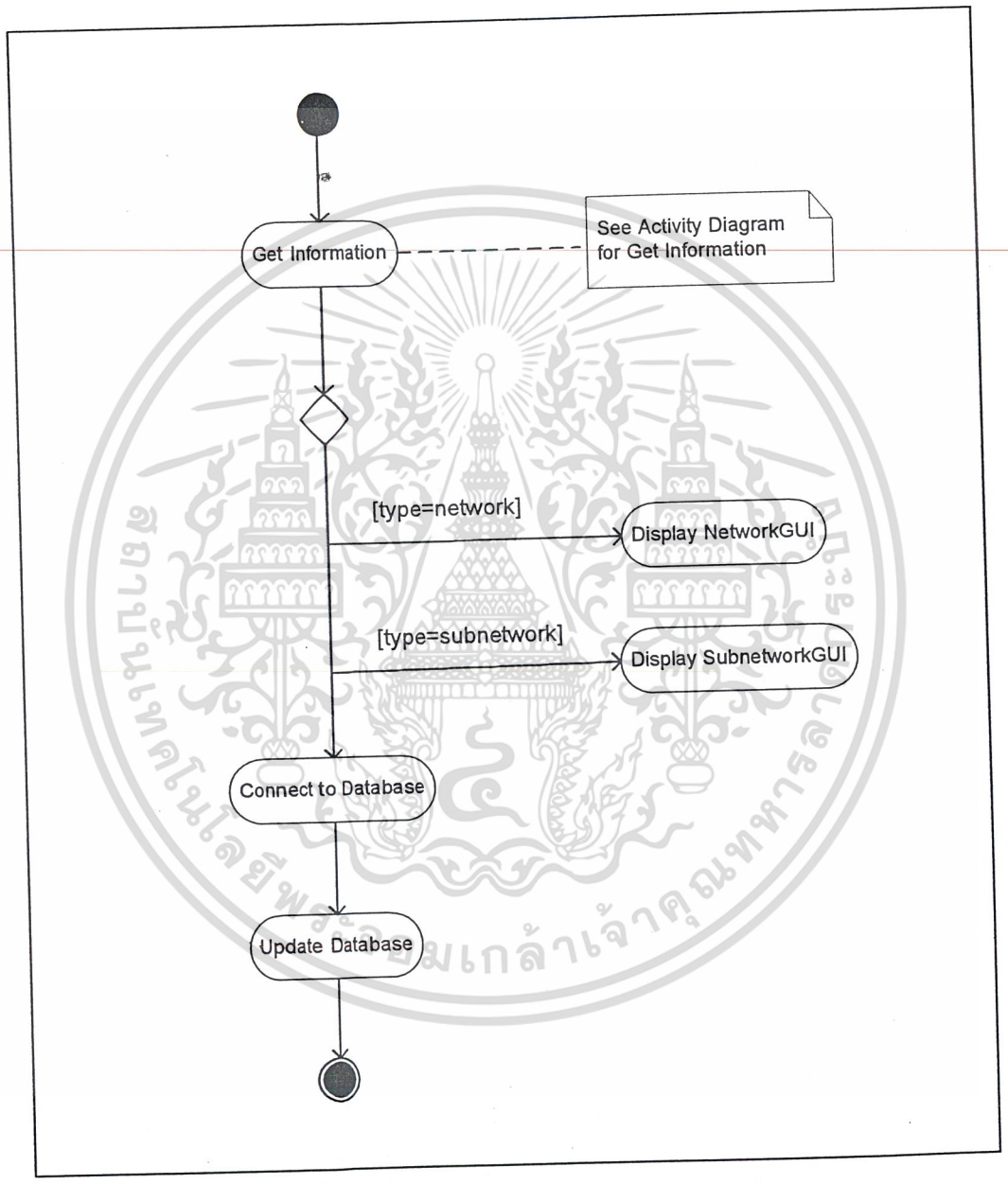
Display Network



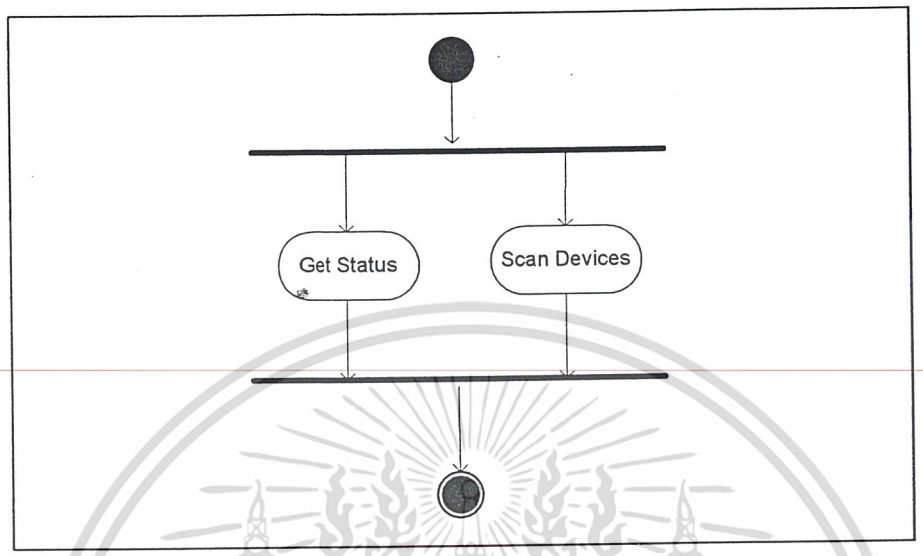
Display Subnetwork



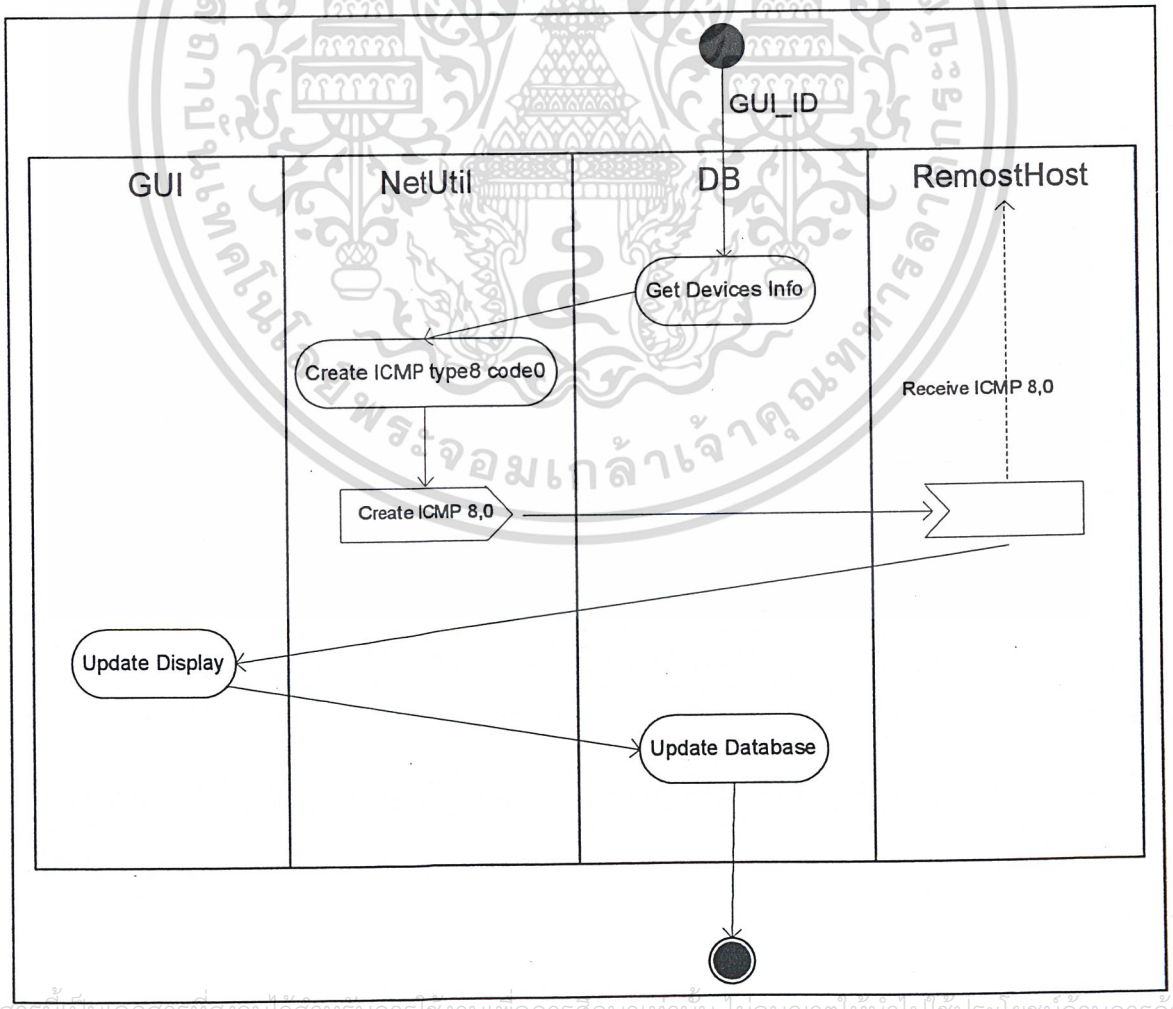
Display Status



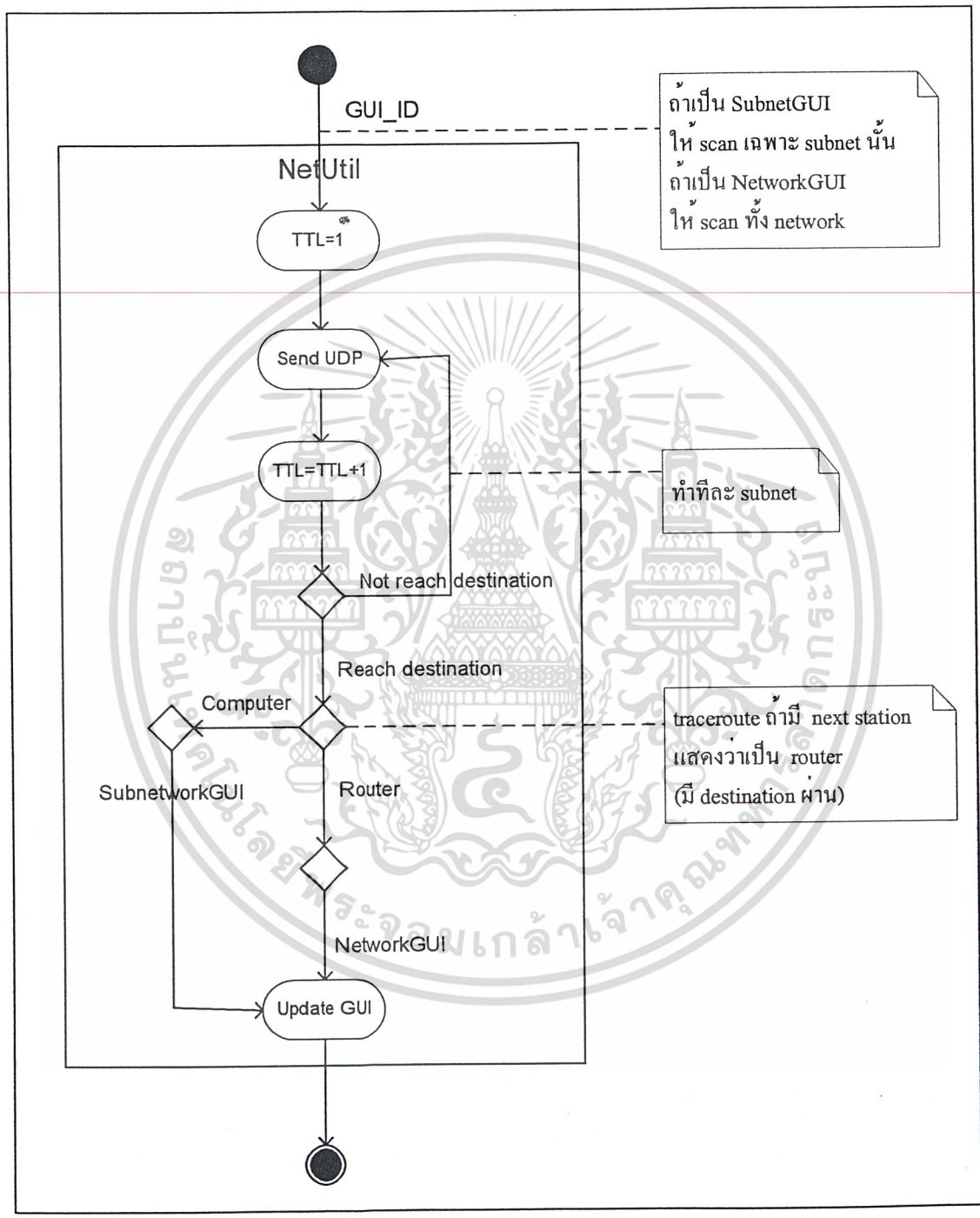
Get Information



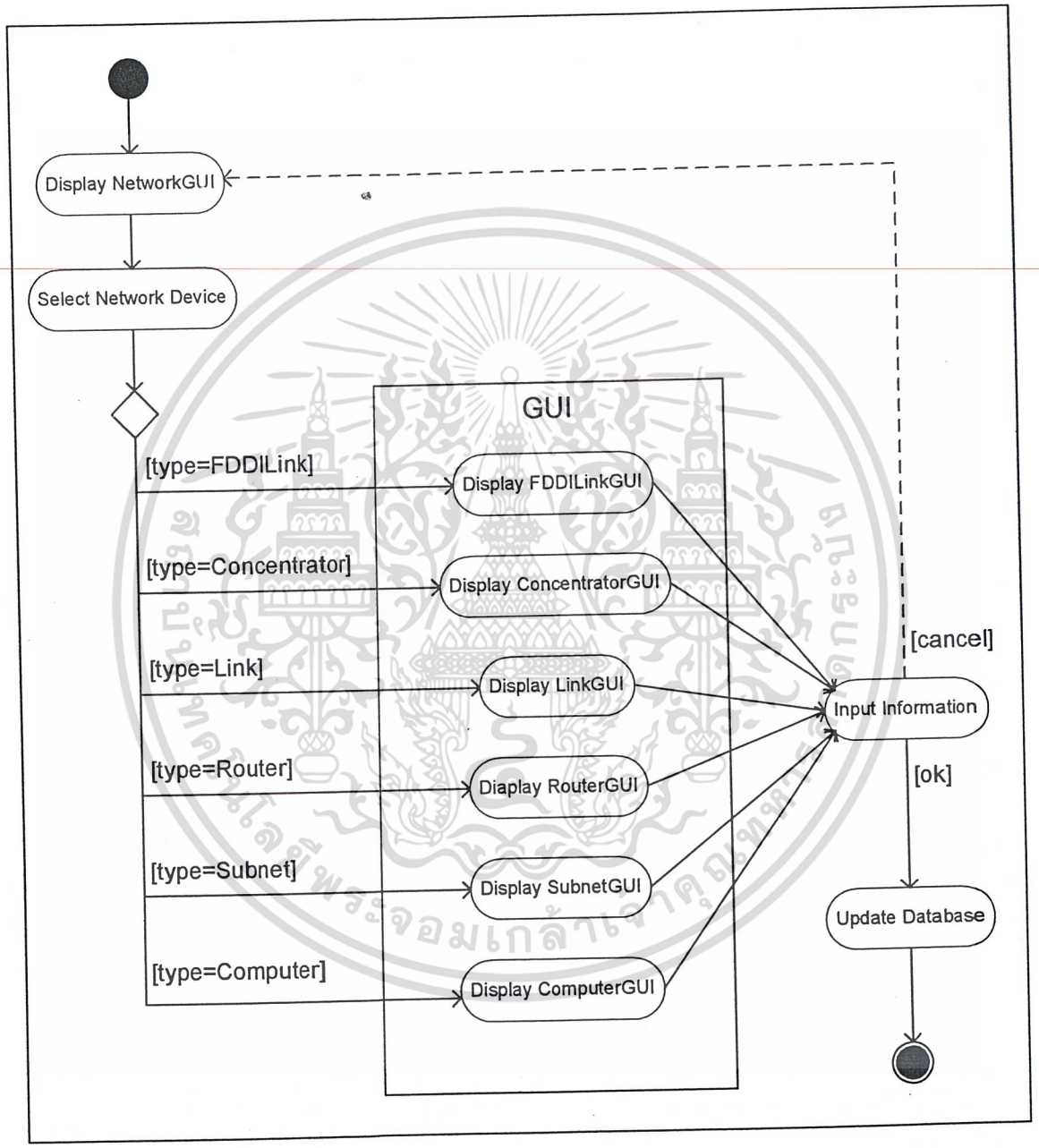
Get Status



Scan Devices

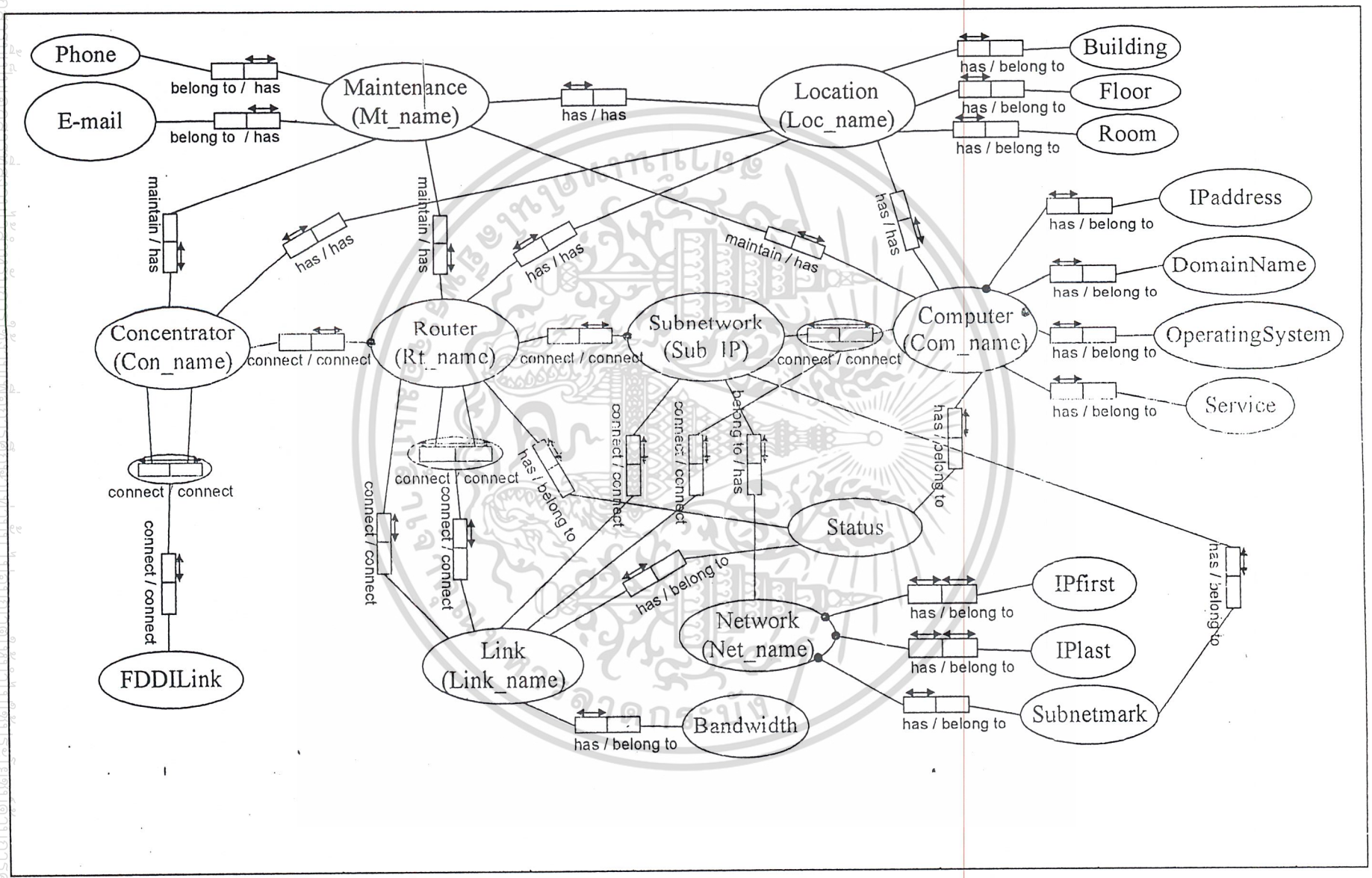


Collect Information

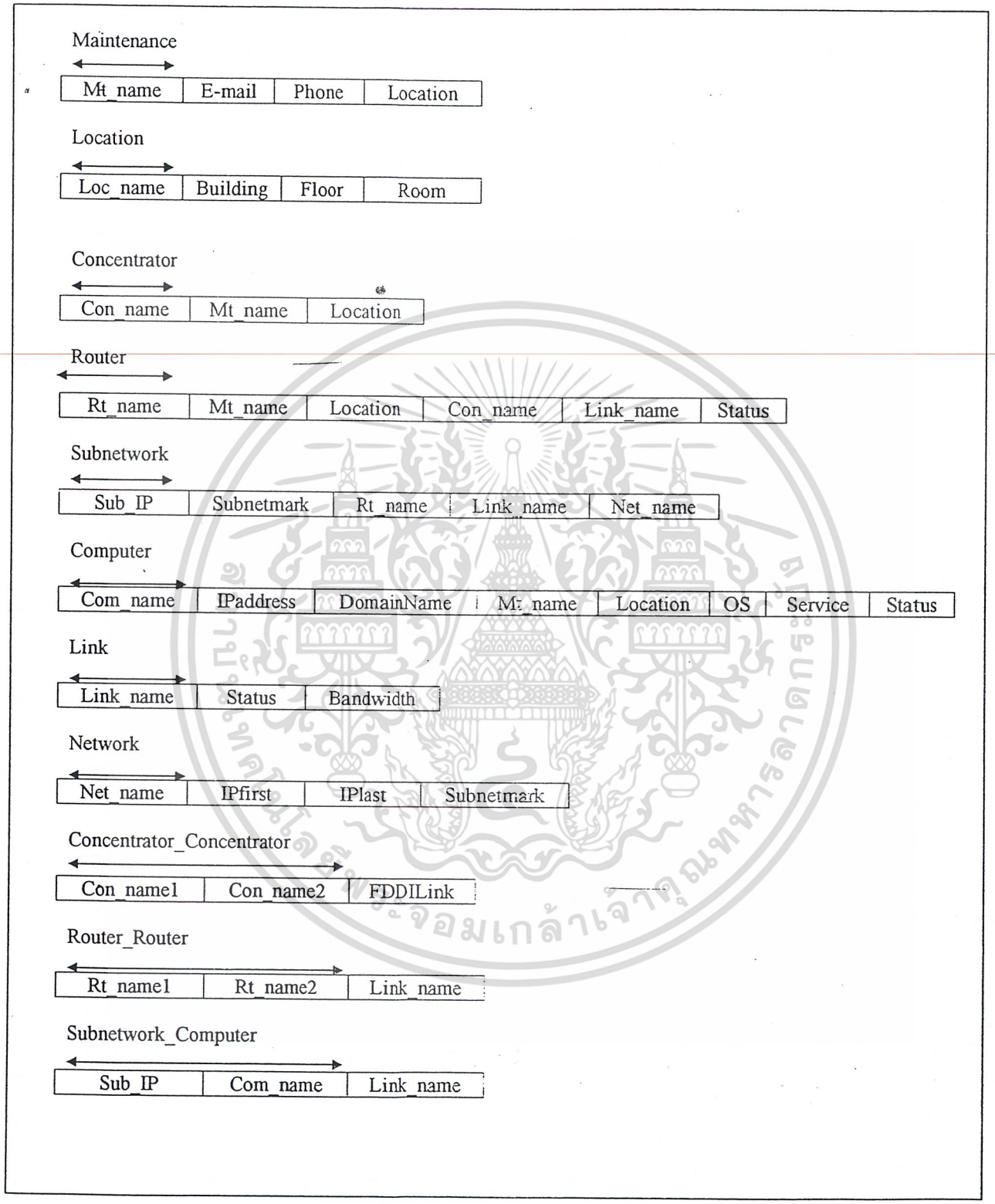


รูปที่ 3.5 แสดงแผนภาพ Activity Diagram

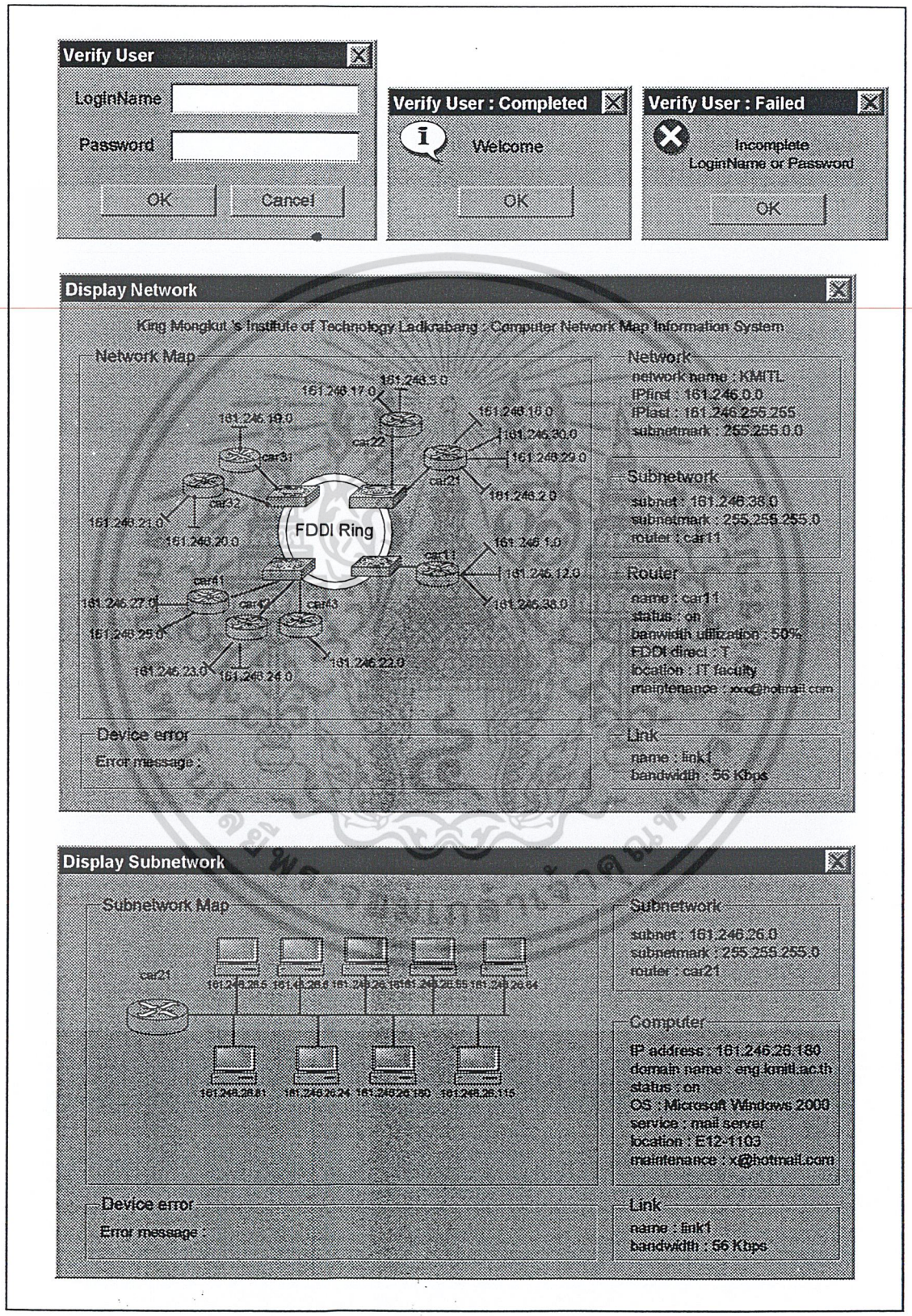
Database Design : NIAM



Relational Database



Graphical User Interface



รูปที่ 3.8 แสดงแผนภาพ Graphical User Interface

บทที่ 4

การทดลอง

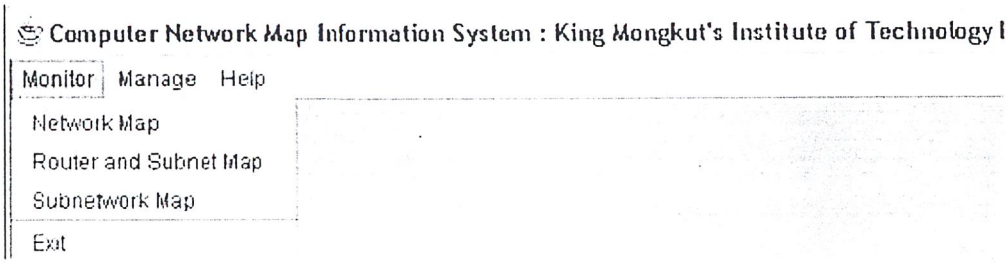
สืบเนื่องจากในเทอมที่ผ่านมาผู้จัดทำได้ทำการออกแบบระบบด้วย UML และทำการรวบรวมข้อมูลบางส่วนเอาไว้ และในเทอมนี้ได้ทำการจัดสร้างระบบสารสนเทศเพื่อจัดเก็บข้อมูลของระบบเครือข่ายคอมพิวเตอร์ของสถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งระบบสารสนเทศฯ ที่จัดสร้างขึ้นนั้นสามารถแสดงผลของข้อมูลในแบบกราฟฟิกซึ่งใช้ภาษาจาวาในการพัฒนา โดยระบบที่สร้างขึ้นนั้นมีคุณสมบัติและความ สามารถหลายประการ ดังเช่น สามารถที่จะทำการแสดงผลสถานะของอุปกรณ์เครือข่าย, ทำการรวบรวมข้อมูลทางด้านเครือข่ายต่างๆ เอาไว้, ทำการประมวลผลและจัดเก็บได้ข้อมูลเหล่านั้นได้อย่างเหมาะสมความต้องการของผู้ดูแลระบบเครือข่ายต้องการ ซึ่งช่วยให้การทำงานของผู้ดูแลระบบเครือข่ายเป็นไปอย่างประสิทธิภาพและเหมาะสม กล่าวคือระบบที่จัดสร้างนั้นมีความสามารถและคุณสมบัติดังต่อไปนี้

ระบบสารสนเทศแผนที่เครือข่ายคอมพิวเตอร์ที่สร้างขึ้นนั้นมีคุณสมบัติและความสามารถข้อมูลต่าง ๆ อันได้แก่

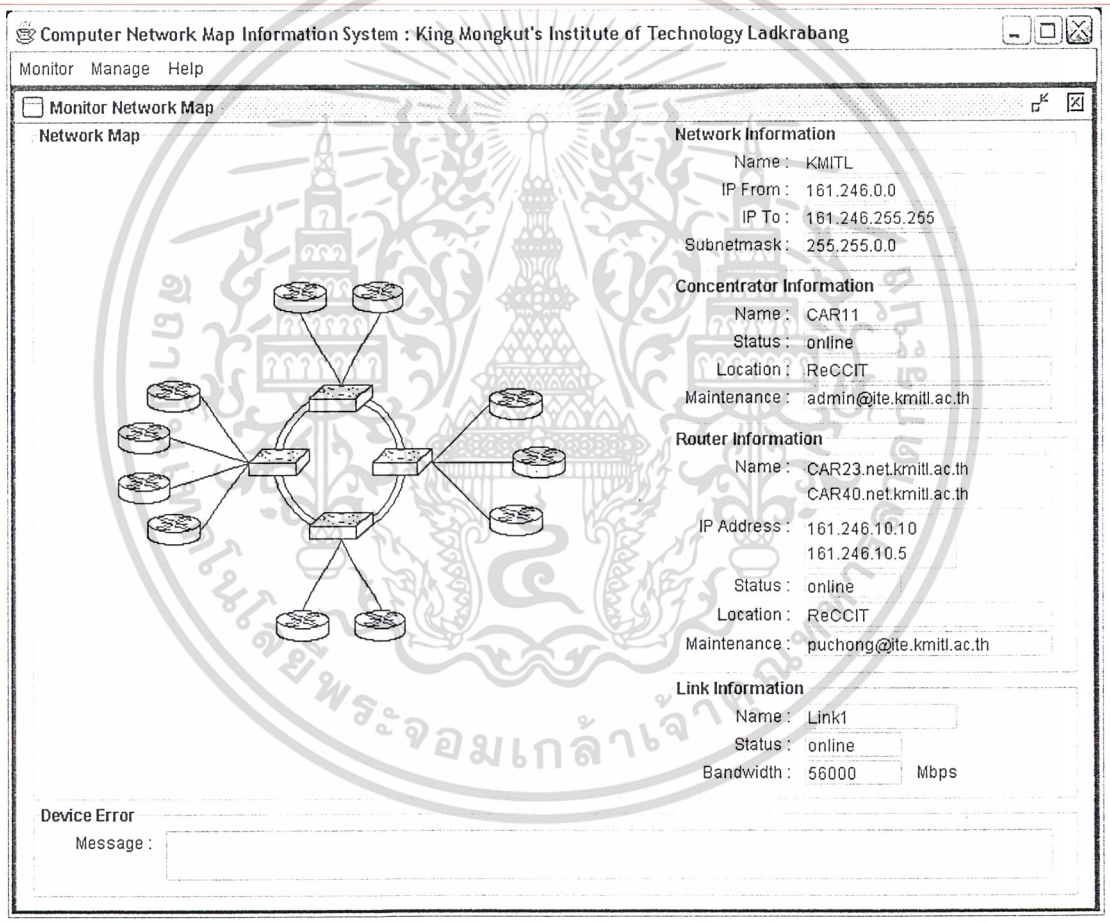
4.1 แสดงการเชื่อมต่อของอุปกรณ์หลัก ในระบบของระบบ

จากการจัดสร้างระบบสารสนเทศฯ ขึ้นมานั้น ระบบสามารถที่จะทำการแสดงผลการเชื่อมต่อของอุปกรณ์แสดงได้ดังภาพที่ รูปที่ 4.1 Monitor Network Map ซึ่งเป็นการแสดงผลการจำลองโครงสร้างเครือข่ายหลักของสถาบันฯ อันประกอบด้วยอุปกรณ์และแสดงการเชื่อมต่อได้แก่ FDDI ring, concentrators, routers, links โดยเครือข่ายของสถาบันฯ จะประกอบไปด้วย concentrater ซึ่งเป็นโหนดหลักอยู่ 4 โหนด และ แต่ละโหนดประกอบด้วยอุปกรณ์ router เชื่อมต่อด้วย link

เมื่อต้องการที่จะเรียกดูผลแสดงการเชื่อมต่อของอุปกรณ์หลักในระบบ โดยจะทำการเลือกที่ Monitor ที่รายการหลักดังรูปที่ 4.1ก จากนั้นเลือกไปที่ Network Map ระบบจะทำการแสดงหน้าต่างซึ่งแสดงอุปกรณ์หลักดังรูปที่ 4.1ข



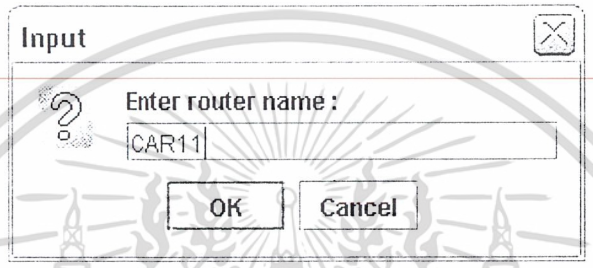
รูปที่ 4.1ก แสดงรายการหลัก



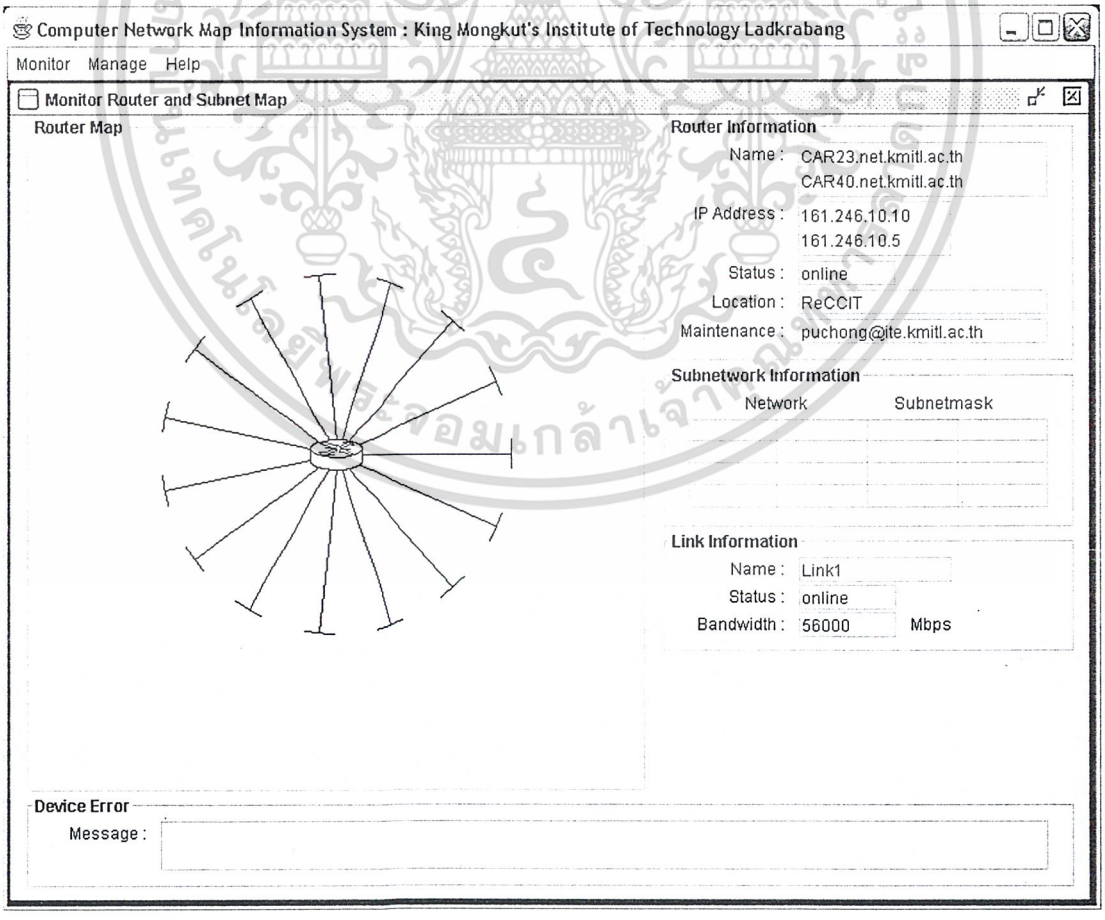
รูปที่ 4.1ข แสดงการเชื่อมต่อของอุปกรณ์หลัก

4.2 สามารถแสดงการเชื่อมต่อของอุปกรณ์ต่างๆ ในระบบ

จากระบบที่จัดสร้างขึ้นนั้น สามารถแสดงการเชื่อมของอุปกรณ์ต่างๆ เช่น เมื่อทำการเลือกที่รายการ ที่ Monitor แล้วทำการเลือก Router and Subnet Map โดยจะแสดงผลการเชื่อมต่อของอุปกรณ์ router ไปยัง subnetwork ต่างๆ ดังรูปที่ 4.2ข จากรูปที่ 4.2ข นั้นยังแสดงรายละเอียดข้อมูลต่างๆ ได้แก่ ข้อมูลของ router ,ข้อมูลของ subnetwork

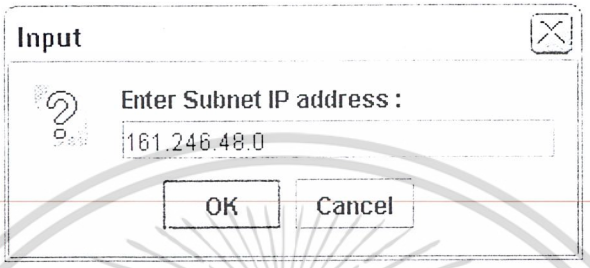


รูปที่ 4.2ก แสดงการป้อน Input ที่ต้องการ เช่น CAR11

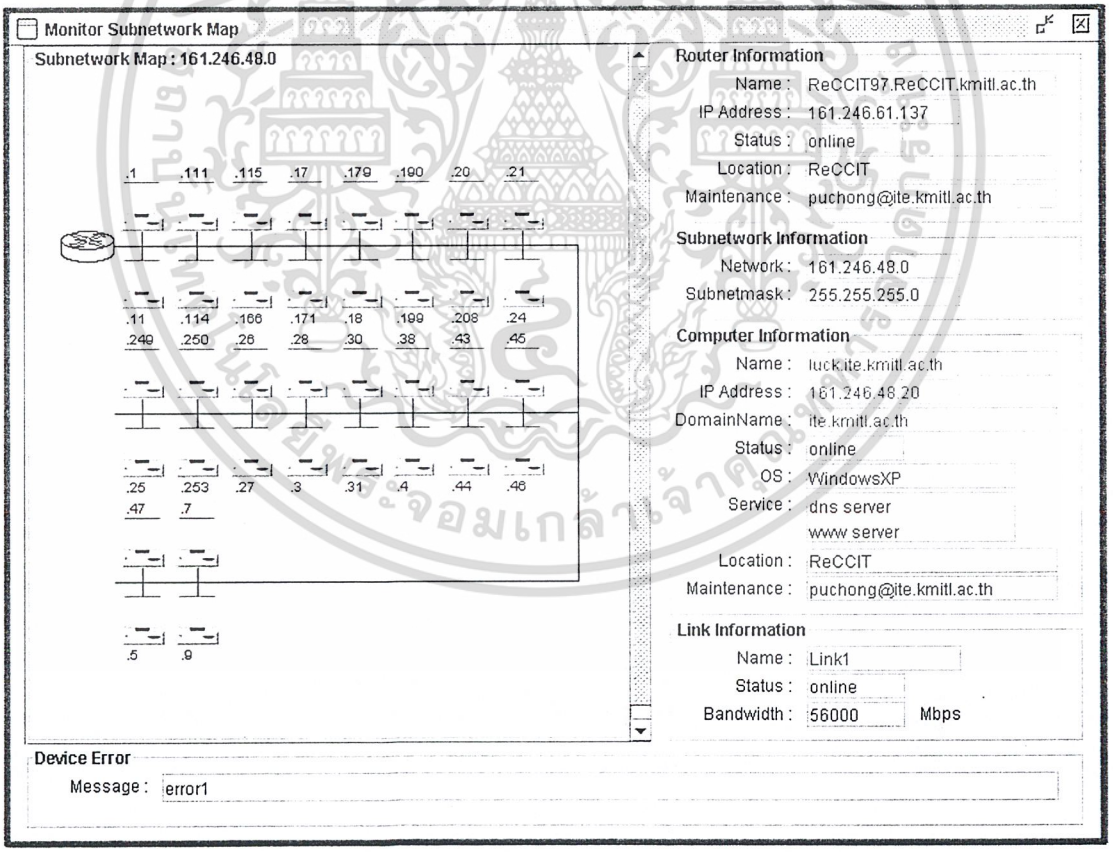


รูปที่ 4.2ข แสดงหน้าต่างของการเชื่อมต่อ Router and Subnet

นอกจากนี้ เมื่อเลือกไปที่ Monitor เลือก Subnetwork แล้วป้อน input ของ Subnetwork ที่ต้องการแสดงผล เช่น ป้อน 161.246.48.0 ดังรูปที่ 4.3ก จากนั้นระบบจะแสดงผลการเชื่อมต่อของ router และ computer ได้ดังรูปที่ 4.3ข



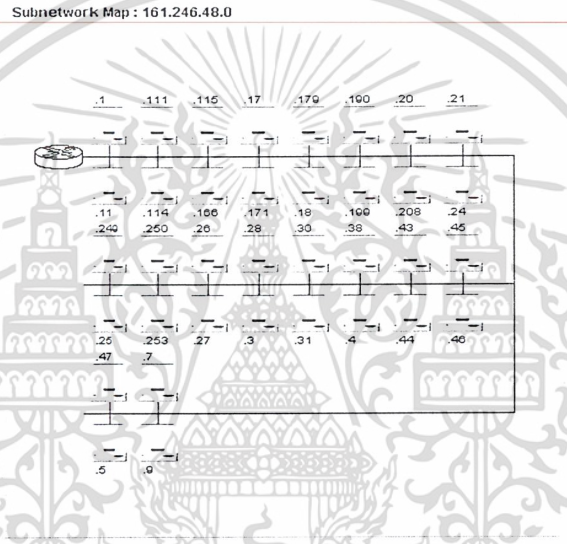
รูปที่ 4.3ก แสดงการป้อน input ในการเลือก subnet ที่ต้องการ



รูปที่ 4.3ข แสดงการเชื่อมต่อของอุปกรณ์ Router และ Computer

4.3 แสดงสถานะของอุปกรณ์ในปัจจุบันว่าเปิดหรือปิด

จากการจัดสร้างระบบสารสนเทศฯ ซึ่งระบบที่ทำการจัดสร้างขึ้นนั้นสามารถแสดงสถานะการเปิดหรือเปิด และสถานะของการใช้งานของอุปกรณ์ที่อยู่ในระบบเน็ทเวิร์กได้ ซึ่งการแสดงสถานะของอุปกรณ์นั้นๆ จะแสดงเป็นรูปไอคอนดังเช่นรูปที่ 4.4 เมื่อทำการเลือกที่หน้ารายการแล้วเลือก Monitor จากนั้น ป้อนข้อมูลของ subnetwork ที่ต้องการให้แสดงผล เช่น ป้อน 161.246.48.1 (ดังรูปที่ 4.3ก) จากนั้นระบบจะทำการแสดงผลดังรูปที่ 4.4



รูปที่ 4.4 แสดงภาพ Subnetwork Information

จากรูปที่ 4.4 เป็นการแสดงสถานะของการใช้งานได้ของอุปกรณ์ router และแสดงสถานะของเครื่องคอมพิวเตอร์ว่า เปิด หรือ ปิดอยู่ ซึ่งจะทำการเช็คไปยังเน็ทเวิร์กเพื่อทำการตรวจสอบ โดยทำการแสดงจำนวนอุปกรณ์ทั้งหมดที่ใช้งานอยู่ ณ ปัจจุบัน

4.4 แสดงหมายเลย IP address ของอุปกรณ์ต่างๆ ในระบบ

สามารถแสดง IP address ซึ่งข้อมูลที่ได้นั้นได้มาจากขั้นตอนการตรวจสอบระบบ และจากข้อมูลจากสถาบัน โดยอุปกรณ์ที่สามารถระบุ IP address ได้แก่อุปกรณ์ router และ computer ดังรูปที่ 4.5ก, 4.5ข และ 4.5ค

IP Address : 161.246.10.10
161.246.10.5

รูปที่ 4.5ก แสดง IP address

Router Information

Name : CAR23.net.kmitl.ac.th
CAR40.net.kmitl.ac.th

IP Address : 161.246.10.10
161.246.10.5

รูปที่ 4.5ข แสดง IP ของ Router

Network Information

Name : KMITL
IP From : 161.246.0.0
IP To : 161.246.255.255
Subnetmask : 255.255.0.0

รูปที่ 4.5ค แสดงช่วงของ IP address ของเครือข่ายสถาบันฯ

4.5 แสดงบริการต่างๆ ที่เครื่องคอมพิวเตอร์เครื่องหนึ่งให้บริการหรือใช้บริการอยู่

แสดงบริการต่าง ๆ ที่เครื่องคอมพิวเตอร์เครื่องหนึ่งให้บริการอยู่ หรือใช้บริการอยู่ โดยจะตรวจเช็คไปยังเครือข่ายเน็ทเวิร์กทั้งเครือข่าย สามารถทราบรายละเอียดของข้อมูลต่างๆ ของเครื่องคอมพิวเตอร์ ได้แก่ ชื่อของเครื่อง, IP address , Domain name, สถานะของเครื่อง, ระบบปฏิบัติการของเครื่อง, แสดงบริการต่างๆ ที่เครื่องให้บริการอยู่ หรือใช้บริการอยู่ ซึ่งการแสดงข้อมูลในส่วนนี้ จะอยู่ที่หน้าต่าง Subnetwork Map ดังรูปที่ 4.6

Computer Information

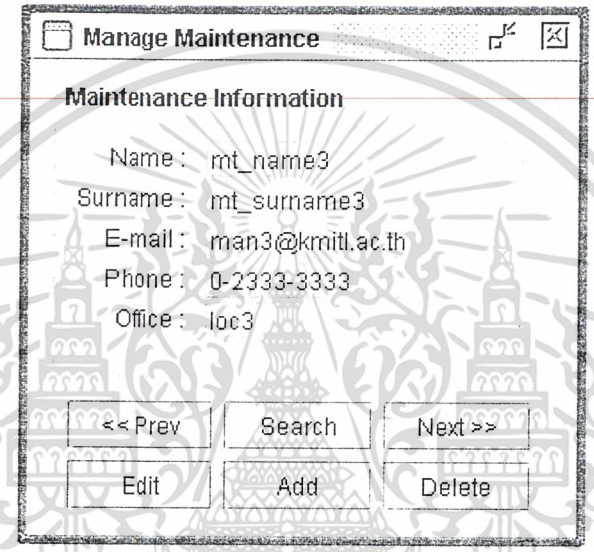
Name :	luck.ite.kmitl.ac.th
IP Address :	161.246.48.20
DomainName :	ite.kmitl.ac.th
Status :	online
OS :	WindowsXP
Service :	dns server
	www server
Location :	ReCCIT
Maintenance :	puchong@ite.kmitl.ac.th

รูปที่ 4.6 แสดงบริการต่าง ๆ ที่เครื่องคอมพิวเตอร์เครื่องหนึ่งให้บริการ

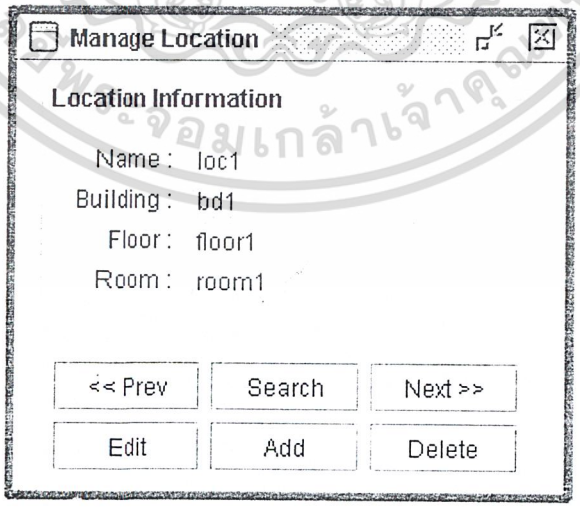
4.6 แสดงสถานที่ตั้งและเจ้าหน้าที่ผู้ดูแลอุปกรณ์

จากโปรแกรมที่ได้พัฒนาขึ้นนั้น สามารถที่ดูรายละเอียดเกี่ยวกับที่สถานที่ตั้งและผู้ดูแลอุปกรณ์ที่ทำการจัดเก็บในฐานข้อมูลได้โดยตรง

เมื่อคลิ๊กไปที่ หน้ารายการหลักแล้วเลือกหัวข้อ Manage และเลือกที่จะทำแสดงรายการ Maintenance หรือ Location ดังรูปที่ 4.7ก และ 4.7ข ซึ่งสามารถจัดเก็บและจัดการข้อมูลได้โดยตรง



รูปที่ 4.7ก แสดงหน้าต่างผู้ดูแลอุปกรณ์



รูปที่ 4.7ข แสดงหน้าที่ตั้งของอุปกรณ์

บทที่ 5
สรุปผลการทดลอง
วิจารณ์ปัญหา และแนวทางการพัฒนาในอนาคต

5.1 สรุปผลการทดลอง

ในการจัดสร้างระบบสารสนเทศแผนที่เครือข่ายคอมพิวเตอร์ สามารถทำการจัดเก็บรวบรวมข้อมูลของเครือข่ายคอมพิวเตอร์ที่มีอยู่อย่างมาก และข้อมูลมีการเปลี่ยนแปลงอยู่ตลอดเวลาตามการใช้งาน ระบบที่สร้างขึ้นสามารถทำการจัดเก็บข้อมูลเหล่านั้น และวิเคราะห์ประมวลผลข้อมูล เพื่อใช้ในการแสดงผลในรูปแบบกราฟฟิก ความสามารถของระบบนั้นอันได้แก่

1. แสดงการเชื่อมต่อของอุปกรณ์หลัก ในระบบของระบบ
2. สามารถแสดงการเชื่อมต่อของอุปกรณ์ต่างๆ ในระบบ
3. แสดงสถานะของอุปกรณ์ในปัจจุบันว่าเปิดหรือปิด
4. แสดงหมายเลย IP address ของอุปกรณ์ต่างๆ ในระบบ
5. แสดงบริการต่างๆ ที่เครื่องคอมพิวเตอร์เครื่องหนึ่งให้บริการหรือใช้บริการอยู่
6. แสดงสถานที่ตั้งและเจ้าหน้าที่ผู้ดูแลอุปกรณ์

ซึ่งรายละเอียดต่างๆ ได้กล่าวเอาไว้ในหัวข้อการทดลอง ด้วยคุณสมบัติเหล่านี้ของระบบสารสนเทศแผนที่เครือข่ายคอมพิวเตอร์ นั้นสามารถช่วยการทำงานของผู้ดูแลระบบเครือข่ายคอมพิวเตอร์ของสถาบัน ฯ ได้อย่างมีประสิทธิภาพและเหมาะสมตามต้องการระดับหนึ่ง

5.2 วิจารณ์ปัญหาที่เกิดขึ้นจากการทดลอง

จากการสร้างระบบสารสนเทศ ฯ นั้น ในส่วนของปัญหาที่เกิดขึ้นตั้งแต่เริ่มทำการสร้างและตราบจบเสร็จสิ้นนั้น นั้นมีหลายประการแต่สามารถสรุปปัญหาได้แก่ ปัญหาที่เกิดจากความล่าช้าของการเก็บรวมข้อมูล เนื่องจากข้อมูลทางด้านเน็ทเวิร์กมีอยู่จำนวนมากมายดังนั้นต้องใช้เวลาในการจัดการและวิเคราะห์คุณสมบัติของข้อมูลเหล่านั้นเพื่อใช้ในการประมวลผล

5.2 แนวทางในการพัฒนาต่อ

ระบบสารสนเทศฯ ที่สร้างขึ้นนั้นสามารถที่จะใช้งานได้ดีในระดับหนึ่ง แต่จะต้องมีการพัฒนาในบางส่วนเพื่อที่ทำให้ระบบสามารถทำงานได้เต็มประสิทธิภาพ

สำหรับแนวทางในการพัฒนาต่อไปในอนาคตนั้นสามารถทำการพัฒนาในเรื่อง

1 . เมื่ออุปกรณ์ทำงานผิดพลาดหรือเกิดการชำรุดเสียหาย ระบบจะทำการแจ้งเตือนไปยังผู้ดูแลระบบ เพื่อให้ผู้ดูแลระบบได้ทราบถึงความผิดปกติของอุปกรณ์นั้นๆ อย่างทันท่วงที ซึ่งตัวอย่างของการแจ้งเตือนดังเช่น การส่งข้อความสั้นๆ ไปยังโทรศัพท์มือถือของผู้ดูแลระบบ , ส่ง e – mail เพื่อแจ้งเตือน

2. ในเรื่องของการเพิ่มประสิทธิภาพในประมวลผลและวิเคราะห์ข้อมูลที่ได้รับมาจากระบบเครือข่ายการตรวจเช็คสถานะและตรวจสอบแอปพลิเคชันที่ใช้ในคอมพิวเตอร์ที่ทำการเปิดใช้งานอยู่ เพิ่มประสิทธ์ภาพของระบบให้สามารถทำงาน ณ เวลาปัจจุบันดียิ่งขึ้น

บรรณานุกรม

1. อภิเนตร อูนากูล, Object Oriented Analysis And Design, วศ.สจล. 162 ภาควิศวกรรมวิชาคอมพิวเตอร์ คณะวิศวกรรมศาตร์ สถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543
2. สุรศักดิ์ สงวนพงษ์ , สถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพี , ภาควิศวกรรมวิชาคอมพิวเตอร์ ,คณะวิศวกรรมศาตร์ , มหาวิทยาลัยเกษตรศาตร์, 2543
3. Joseph Schmuller , SAMS Teach Yourself UML in 24 Hours , Techmedia, 1999
4. H.M. Deitel Deitel & Association,Inc, D.J. Deitel Deitel & Association,Inc, How to java program third edition,1999